SGMNet: Learning Rotation-Invariant Point Cloud Representations via Sorted Gram Matrix

Jianyun Xu, Xin Tang, Yushi Zhu, Jie Sun, Shiliang Pu*
Hikvision Research Institute
{xujianyun, tangxin10, zhuyushi, sunjie, pushiliang.hri}@hikvision.com

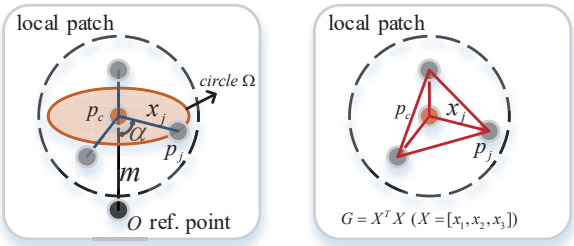
Abstract

*Recently, various works that attempted to introduce rotation invariance to point cloud analysis have devised point-pair features, such as angles and distances. In these methods, however, the point-pair is only comprised of the center point and its adjacent points in a vicinity, which may bring information loss to the local feature representation. In this paper, we instead connect each point densely with all other points in a local neighborhood to compose the point-pairs. Specifically, we present a simple but effective local feature representation, called **s**orted **G**ram **m**atrix(SGM), which is not only invariant to arbitrary rotations, but also models the pair-wise relationship of all the points in a neighborhood. In more detail, we utilize vector inner product to model distance- and angle-information between two points, and in a local patch it naturally forms a Gram matrix. In order to guarantee permutation invariance, we sort the correlation value in Gram matrix for each point, therefore this geometric feature names sorted Gram matrix. Furthermore, we mathematically prove that the Gram matrix is rotation-invariant and sufficient to model the inherent structure of a point cloud patch. We then use SGM as features in convolution, which can be readily integrated as a drop-in module into any point-based networks. Finally, we evaluated the proposed method on two widely used datasets, and it outperforms previous state-of-the-arts on both shape classification and part segmentation tasks by a large margin.*

1. Introduction

In the past few years, the development of 3D computer vision has flourished due to its wide applications, such as robotics, AR/VR and self-driving. Point cloud, the most common format of 3D data, has attracted more and more attention, and plenty of works [19, 20, 1, 25, 13, 26, 24] have been proposed to extract 3D features directly on points.

*Corresponding author. This work was supported by National Key R&D Program of China (Grant No.2020AAA010400X).



(a) Simplified PPFs (b) Dense-connected PPFs

Figure 1. Illustration of simplified PPFs and our improved dense-connected version. In figure(a), the PPFs are constructed between center point p_c and neighbor point p_j by distance $||x_j||(x_j = p_j - p_c)$ and angle $\angle(m, x_j)(m = Op_c$, O is a reference point, such as mass center), which cannot fully constraint p_j, letting it freely positioned in the circle Ω, thus leads to information loss. In figure(b), we densely connect every point-pair exhaustively and compute each pair's distance and angle information by an inner product which, forms a Gram matrix, leading to a fixed local structure.

Compared to dense images, point clouds are irregular and unstructured, thus it is necessary to design permutation invariant feature extractors, which is handled by PointNet [19] using symmetric functions. Besides that, spatial transformation, including translation and rotation, is generally common in 3D world. Therefore, the learned representation of the point cloud should also be invariant to translation and rotation [19]. While translation-invariance can be accomplished perfectly by using weight sharing as 2D convolution does in an image, rotation-invariance in 3D point cloud has not been well resolved due to its spacial complexity.

In essence, rotation-invariance is crucial in some specific tasks, such as classification, part segmentation, matching and shape retrieval, as in these tasks, the objects or the whole scene can be arbitrarily rotated. Hence, it is necessary to ensure that the learned point cloud feature is rotation-invariant, and in particular, the local patch descrip-

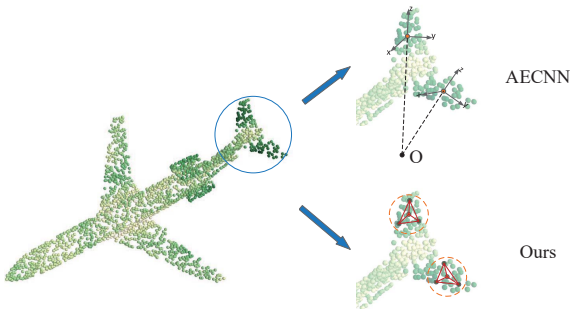


Figure 2. Illustration of rotation-invariant local feature representation. Upper figure: it shows the local feature extraction of AECNN [31]. For the two emages of this airplane, though sharing the same structure, their local feature is discriminative due to different LRF construction, thus the local feature of AECNN is not rotation invariant. Note that, this is why AECNN needs to align features in different LRFs. Lower figure: it shows our proposed local feature descriptor, which is strictly rotation-invariant for two local patches as long as their inherent structures keep the same.

tor should also be rotation invariant, i.e., two point cloud patches that share the same inherent structure should have the same feature representation.

To cope with the rotation issue, a straightforward method is to feed the model with great amounts of rotation augmented data. For further rotation robustness, some [19, 25] used the STN [10] to transform input to canonical space. Nevertheless, all these methods rely heavily on data augmentation and is insufficient to ensure generalizability to unseen rotations. Besides, these augmentation-based methods are obliged to extract features from an extremely huge input space, leading to higher model complexity. On the contrary, rotation-invariant methods have lower model complexity.

Studies on rotation-invariant deep learning have already been proposed in the past few years. One stream of methods [11, 28, 31, 33] tried to construct local reference frames(LRFs) and transfer coordinates from global to local frame, thus are invariant to global rotations. AECNN [31] constructs the LRF using the global origin point as reference, making its LRFs vary even for the same two local patches, as depicted in Figure 2, so its local feature descriptor is **not** rotation-invariant, limiting its application to other tasks, such as object detection and matching. Other methods [11, 28] utilize priciple component analysis (PCA) to construct the LRF, sufferring from the ambiguity of eigen vectors.

Another intuitive way is to utilize naturally rotation-invariant features, e.g. angles and distances, as the network input. Drost et al. [7] first proposes point-pair features (PPFs) to build rotation-invariant network, by constructing distance and angles between two points and their normal vectors. Later methods [32, 23] simplified the PPFs by removing the computationally costly normal estimation. However, this leads to much information loss for local representation, as depicted in Figure 1(a). ClusterNet [3] defined a plane using original point as reference and sort the projected angles among neighbors, which fully constrained the local patch, but loses the rotation-invariance for its local feature representation similar to AECNN [31]. For all these PPF-based methods, although they help the network generalize to unseen rotations, their performance is relatively lower than LRF-based methods.

For simplicity and generalizability, we follow the track of utilizing point-pair features, and improve it by connecting every pair of points exhaustively in a neighborhood to obtain a complete local geometric representation. To this end, we use the vector inner product between all point pairs to model the distance and angle information, which naturally forms a Gram matrix in a neighborhood. In order to guarantee permutation-invariant, we sort the correlation value in the Gram matrix for each point, so this geometric feature names sorted Gram matrix. We use the SGM themselves as features in the convolution, which can be readily implemented as a drop-in module in any point-based network.

Main contributions are summarized as follows:

- We propose a simple and effective local feature representation, named SGM, which not only keeps invariant to arbitrary rotations, but also models the pairwise relationship of all the points in a neighborhood.
- We pack the SGM as a drop-in module, and integrate it into many popular point-based networks, showing the robustness, efficiency and generalizability of the SGM.
- We illustrate that our proposed method is invariant to rotation and achieves state-of-the-art results in both point cloud shape classification and part segmentation tasks.

2. Related Works

2.1. Deep Learning on Point Clouds

In recent years, 3D point cloud processing using deep neural networks has drawn much research interest and showed superior performance in multiple 3D tasks [19]. 3D Shapenet [27] and Voxnet [16] introduced 3D convolutional neural networks to volumetric data due to its similarity to image data. However, 3D CNN results in large memory consumption and much calculation is wasted on sparse voxels. PointNet [19] is the pioneer work to use raw point cloud as input without transferring to regular 3D data. It maps the original 3D coordinates to high-dimensional features with MLPs, followed by a permutation-invariant pooling operation to aggregate individual features to global fea-

tures. However, PointNet neglects the crucial role of local shape descriptors in various high-level 3D understanding tasks. Several follow ups attempt to learn local features in different ways. PointNet++ [20] used PointNet as local feature descriptor and formed a hierarchical structure. DGCNN [25] utilized graph and incorporated edge information. Other works [12, 13, 26, 24] have made some exploration on adapting convolution operator to point cloud learning. In this study, we aim to learn DNNs with rotation invariance property.

2.2. Rotation-robust 3D Networks

One intuitive way to achieve rotation robustness is to feed the network with great amounts of rotation-augmented data. However, data augmentation cannot ensure generalization to unseen rotations and have higher model complexity.

Some methods [4, 8, 21] project 3D data to a unit sphere and perform convolutions using a spherical harmonic basis. Rasterization in these methods make them not able to maintain consistent performance when rotated, and spherical projection may be difficult to handle complex non-convex objects. [30] proposed Spherical Voxel Convolution to extract point-wise rotation-invariant features at spherical voxel grids. However, it didn't solve the information loss problem and performance drops when testing with arbitrary rotated point cloud.

Some methods resort to geometric features between point pairs [5, 3, 32, 23] in pursuit of inherent rotation-invariant property. PPF-FoldNet [5] leveraged the difference vector between point pairs and relative angle of their normal vectors. Similar to PPF-FoldNet [5], ClusterNet [3] utilized distance and angle to define rotation-invariant features. However, it used origin point as reference point to construct the projection plane and angle, resulting in rotation-variant local features. RIConv [32], on the contrary, used mass center as reference point to construct rotation-invariant features by using simplified point-pair features. These methods could achieve global rotation-invariance in certain degrees but are far from the performance of standard point cloud learning methods in aligned scenarios on basic tasks such as classification and segmentation.

Other methods [31, 11, 33, 28] learned rotation-invariant representations by introducing local reference frames (LRFs) into the networks. However, LRFs are sensitive to point density changes, ambiguous and sometimes non-unique, which limit the representational power and generalization performance to unseen rotations. Moreover, LRF-based methods may generate different features for the same but rotated local patches depending on its LRF definition, as shown in Figure 2, which are deviated from the initial motivation and thus not suitable for tasks like scene segmentation, object detection and point cloud matching, where locally rotated patches exist.

One common problem of all the previous works is that they either rely on extra information or very complicated design. Our work is more intuitive, straightforward and easy to integrate to various point-based networks. Furthermore, methods above are not all rotation-invariant with regard to local features, such as ClusterNet [3] and AECNN [31]. Their global features are rotation invariant while local feature are *not*, which constrains their application in tasks like matching and object detection.

3. Methodology

In this section, we firstly state the rotation-invariance problem, then revisit the concepts of point-pair features and Gram matrix, followed by the core contribution of this paper, i.e., sorted Gram matrix. Finally, the network architecture is presented.

3.1. Problem Statement

We study the problem of rotation-invariant local descriptors for point clouds. Given a point cloud patch with K points, denoted as $X = [x_1, ..., x_K]$, where each $x_i (i = 1, ..., K)$ contains the 3D coordinates of a point in Euclidean space, thus $X \in \mathbb{R}^{3\times K}$. Note that, the local patch discussed in this paper is already translated by the center of this patch. For any given $R \in SO(3)$ (*special orthogonal group*), we need to find a mapping $\mathcal{F} : \mathbb{R}^{3\times K} \mapsto \mathbb{R}^{C\times K}, C \in \mathbb{N}^+$, in which identical point clouds in different orientations should be unified as a unique and consistent representation. Mathematically, it can be formulated as follows:

$$\mathcal{F}(X) = \mathcal{F}(RX) \tag{1}$$

where the $\mathcal{F}(\cdot)$ is the rotation-invariant operation we pursuit, and the $\mathcal{F}(X)$ is the rotation-invariant descriptor of X.

3.2. Preliminaries

Point-Pair Features(PPFs). As illustrated in [7, 2, 9], PPFs are antisymmetric 4D descriptors of a pair of oriented 3D points p_c and p_j, constructed as:

$$(||x_j||_2, \quad \angle(n_c, x_j), \quad \angle(n_j, x_j), \quad \angle(n_c, n_j)) \tag{2}$$

where x_j denotes the difference vector between points computed as $x_j = p_j - p_c$, n_c and n_j are the surface normals at p_c and p_j. $||\cdot||$ is the Euclidean distance and $\angle$ is the angle operator.

In our case, the point cloud comes as coordinates in 3D space without surface normals, and sometimes it is difficult to compute normals especially for sparse point cloud. In the consideration of simplicity and generalizability, we

simplify the PPFs, similar to [32], to a short representation without normal vectors, formulated as:

$$(||x_j||_2, \quad \angle(m, x_j)) \tag{3}$$

where x_j still denotes the difference vector between p_c and p_j, m is the vector between reference point O and p_c, as depicted in Figure 1(a). This simplified feature is invariant under arbitrary rotation as the distances and angles are preserved between every pair of points, which has been proved in many previous works[2, 6, 3, 32].

We use this simplified PPFs as baseline to explore the rotation-invariant representation for a local point set, and discover that in this way only the relationship between center point p_c and its neighbor p_j is established, but the whole geometric structure is not captured, i.e., the inherent relation among neighbors are not constrained, as illustrated in Figure 1(a), thus leading to ambiguity (or in another term, information loss) for a local point cloud description.

This inspired us that the point-pair relation should not only be computed between a central reference and its neighbor points, but take all points in the neighborhood into account, as depicted in Figure 1(b), encouraging us to find a more suitable descriptor.

Gram Matrix. It is intuitive that inner-product between two vectors comprises both distance and angle information. If we compute the inner-product between every pair of points in a point cloud patch, then it naturally forms a Gram matrix. More specifically, for a given point cloud $X \in \mathbb{R}^{3\times K}$, the Gram matrix G of X can be defined as:

$$G(X) = X^T X = \begin{bmatrix} x_1^T x_1 & x_1^T x_2 & \dots & x_1^T x_K \\ x_2^T x_1 & x_2^T x_2 & \dots & x_2^T x_K \\ \vdots & \vdots & \ddots & \vdots \\ x_K^T x_1 & x_K^T x_2 & \dots & x_K^T x_K \end{bmatrix} \tag{4}$$

where each $x_i (i = 1, ..., K)$ contains the 3D coordinates of a point in 3D space.

Obviously, $G(X)$ is a $K \times K$ matrix, and comprises the geometric information of all the point pairs in a compact way. We will first prove that the Gram matrix is rotation-invariant, and then illustrate that this representation is sufficient to model the inherent structure of a point cloud patch.

Theorem 1. *Gram matrix is rotation-invariant, i.e., for* $\forall R \in SO(3), \forall X \in \mathbb{R}^{3\times K} (K \in \mathbb{N}^+)$*, that satisfies*

$$G(X) = G(RX) \tag{5}$$

Proof. For $\forall R \in SO(3), \forall X \in \mathbb{R}^{3\times K}$, according to GM definition in Eq.4,

$$G(RX) = (RX)^T RX = X^T R^T RX = X^T X \tag{6}$$

so $G(RX) = G(X)$, i.e., Gram matrix is rotation-invariant. □

Then we are going to illustrate the Gram matrix is sufficient to model the inherent structure of a point cloud patch.

Lemma 1. *If the Gram matrix of two point clouds are equal, then their geometrical structures are also the same, i.e., one can be represented by the other through a rotation or a reflection.*

Proof. We derived this lemmea from [18]: For two point cloud patches, $X_1, X_2 \in \mathbb{R}^{3\times K}$, $K \in \mathbb{N}^+$, according to [18],

$$min_Q ||X_1 - QX_2||_F \leq \frac{\delta(X_1)}{\sqrt{2}} (1 - \sqrt{1 - \frac{2||G(X_1) - G(X_2)||_F}{a^2(X_1)}}) \tag{7}$$

where Q is a rotation or reflection matrix, $\delta(X_1)$ is the smallest singular value of X_1. Since $G(X_1) = G(X_2)$, it implies $min_Q ||X_1 - QX_2||_F \leq 0$, so $X_1 = QX_2$, which completes the proof. □

3.3. Learning from Sorted Gram Matrix

Sorted Gram Matrix. As mentioned above, the Gram matrix $G(X)$ of a point cloud patch $X \in \mathbb{R}^{3\times K}$ is rotation-invariant and sufficient to model the inherent structure of X. In this section, we intend to use the $G(X)$ itself as the initial feature in a network, however, the $G(X)$ can not be straightly used as it is not permutation-invariant, which is indeed required by point-based networks.

We tackle this problem by sorting the correlation value of Gram matrix for each row in ascending or descending order, and the sorted Gram matrix can be defined as:

$$SGM(X) = f_{sort}(G(X)) = \begin{bmatrix} f_{sort}(X_1) \\ \vdots \\ f_{sort}(X_K) \end{bmatrix} \tag{8}$$

where $f_{sort}(\cdot)$ is the row-wise sorting function, $X_i = [x_i^T x_1, x_i^T x_2, ..., x_i^T x_K] (i = 1, ..., K)$ is the correlation value for each row.

Before implementing the SGM in a netwrok, we illustrate the rotation- and permutation-invariance property of SGM firstly.

Theorem 2 (Rotation Invariance)**.** *For* $\forall X \in \mathbb{R}^{3\times K}$*, the SGM of* X *defined by Eq.8 is rotation-invariant.*

Proof. According to theorem 1, Gram matrix is rotation-invariant, so

$$SGM(RX) = f_{sort}(G(RX)) = f_{sort}(G(X)) \tag{9}$$

so $SGM(RX) = SGM(X)$, i.e., the SGM is rotation-invariant. □

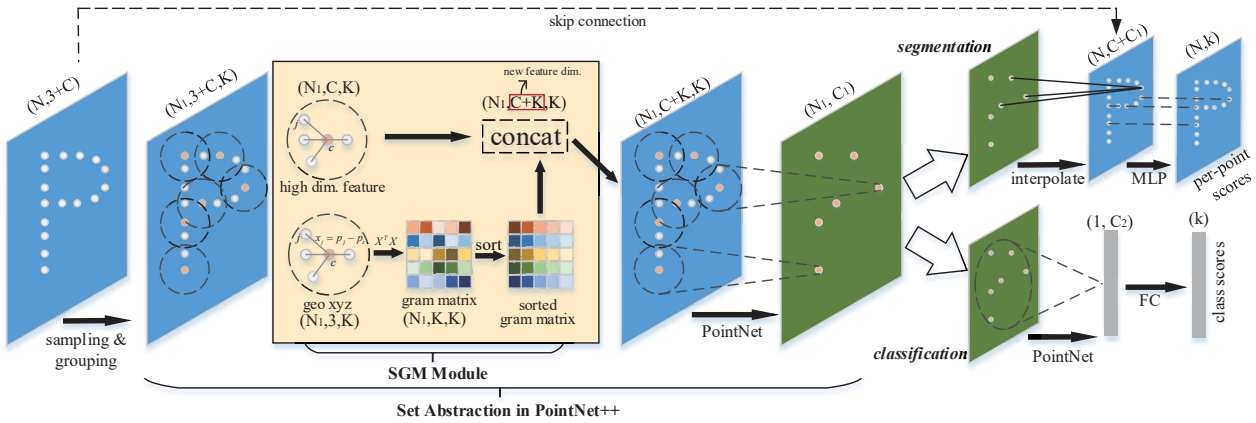


Figure 3. Overview of SGM ensembled in PointNet++. For simplicity, only one set abstraction with one scale is pictured.

Theorem 3 (Permutation Invariance). *For $\forall X \in \mathbb{R}^{3\times K}$, the SGM of X defined by Eq.8 is permutation-invariant after a symmetric function, formulated as:*

$$\mathcal{A}(SGM(XP)) = \mathcal{A}(SGM(X)) \tag{10}$$

where P is a $K \times K$ column permutation matrix, and $\mathcal{A}(\cdot)$ is a symmetric function used for feature aggregation.

Proof. Let $\forall X \in \mathbb{R}^{3\times K}$, for any $K \times K$ column permutation matrix P,

$$\begin{aligned} SGM(XP) &= f_{sort}(G(XP)) \\ &= f_{sort}((XP)^T(XP)) \\ &= f_{sort}(P^T X^T XP) \\ &= f_{sort}(P^T G(X)P) \end{aligned} \tag{11}$$

as f_{sort} is a row-wise sort function and P^T is a row permutation matrix, thus

$$f_{sort}(P^T G(X)P) = P^T f_{sort}(G(X)P) = P^T f_{sort}(G(X)) \tag{12}$$

combine Eq.11, 12,

$$\mathcal{A}(SGM(XP)) = \mathcal{A}(P^T f_{sort}(G(X))) = \mathcal{A}(P^T SGM(X)) \tag{13}$$

as the symmetric function $\mathcal{A}(\cdot)$, e.g. maxpooling, has been proved permutaion-invariant in [19], so

$$\mathcal{A}(P^T SGM(X)) = \mathcal{A}(SGM(X)) \tag{14}$$

namely $\mathcal{A}(SGM(X)$ is permutaion-invariant. □

Learning from Sorted Gram Matrix. In terms of rotation-invariant network, we can use the SGM calculated from points' raw Euclidean coordinates as an initial descriptor for a local point cloud, and concatenate the SGM with their high dimensional features, then feed to subsequent convolution layers, finally aggregate the local features to center point. More formally, without loss of generality, we suppose a local point cloud with K points to have coordinates $X \in \mathbb{R}^{3\times K}$ and features $F \in \mathbb{R}^{C\times K}$, $C \in \mathbb{N}^+$ is the channel number of F (note that in our case the features F is rotaion-invariant, such as semantic features or the previous high-level feature comes from SGM). Then, the learned rotaion- and permutaion-invariant representation f of this local point cloud can be formulated as:

$$f = \delta(\mathcal{A}(MLP(\Psi))) \tag{15}$$

where $\mathcal{A}(\cdot)$ is a symmetric function same as in Eq. 10, δ is an activation funcion,

$$\Psi = \Phi(X, F) = concat(SGM(X), F) \tag{16}$$

We name the $\Phi(\cdot,\cdot)$ as the *SGM Module*, and $\Psi \in \mathbb{R}^{(C+K)\times K}$ is the output of it. $f \in \mathbb{R}^{C_{out}}$ is the final representation of this local point cloud, where C_{out} is the last channel number of the MLP.

Then the *SGM Module* can be readily integrated into any point-based networks for various tasks, which is illustrated in the following section.

3.4. Network Architecture

In this section, we will use the *SGM Module* in hierarchical networks, and illustrate the generality by implement it on classification and segmentation tasks. We leverage a PointNet++ [20] framework to integrate the *SGM Module* into classification and segmentation task, as depicted in Figure 3. Placed in the *Set Abstraction* module of PointNet++, the SGM plays the role of extracting geometrical features, replacing the original xyz coordinates. With this tiny modification the PointNet++ obtains rotation-invariance property, which shows the application potential of *SGM Module*.

4. Experiments

In this section, we first compare our methods with previous state-of-the-art methods on two different tasks: clas-

Method	Inputs	Input size	z/z	$z/SO(3)$	$SO(3)/SO(3)$
PointNet [19]	pc	1024×3	89.2	16.4	75.5
PointNet++ [20]	pc+normal	5000×6	91.8	18.4	77.4
DGCNN [25]	pc	1024×3	**92.2**	20.6	81.1
Spherical CNN [8]	voxel	2×64×64	88.9	76.9	86.9
SFCNN [22]	pc	1024×3	91.4	84.8	**90.1**
PCA-RI [28]	pc	1024×3	88.8	88.8	88.8
L2G-GCN [11]	pc	1024×3	89.5	89.5	89.5
RICNN [32]	pc	1024×3	86.5	86.4	86.4
SRINet [23]	pc	1024×3	87.0	87.0	87.0
ClusterNet [3]	pc	1024×3	87.1	87.1	87.1
SGMNet(ours)	pc	1024×3	90.0	**90.0**	90.0

Table 1. Shape classification results on ModelNet40 dataset. We report the overall classification accuracy (%) under three different train/test settings: (z/z), $(z/SO(3))$, and $(SO(3)/SO(3))$. The results are grouped, from top to bottom, by rotation-sensitive methods, rotation-robust (but not rotation-invariant) methods, LRF-based and PPF-based rotation-invariant methods. Though our method is not the best when compared to rotation-robust method (SFCNN [22]) in z/z and $SO(3)/SO(3)$ cases, but maintain the same performance in all cases. And our method outperforms both LRF- and PPF-based rotation-invariant methods significantly, especially the PPF-based methods that are also based on angles and distances.

sification task on ModelNet40 dataset [27] (Sec. 4.1) and part segmentation task on ShapeNet dataset [29] (Sec. 4.2). Then, in Sec. 4.3, we conduct extensive ablation studies to investigate crucial components of the SGMNet and lead to some promising conclusions.

4.1. Classification

Dataset. We evaluate the classification task on ModelNet40 [27] , which provides 12311 CAD models from 40 object categories. There are 9843 models for training and 2468 models for testing. We use a widely used version provided by PointNet [19], which contains 2048 points in each point clouds. We uniformly sample 1024 points and normalize them to a unit sphere. To note that, only the (x, y, z) coordinates of the sampled points are used.

Implementation Details. The network is trained from scratch with SGD optimizer on a single Tesla V100 GPU. We trained the network for 250 epochs with a batch size of 32, the initial learning rate is 0.001. The cosine annealing learning rate strategy is adopted for the learning rate decay. For the model architecture, we keep the same as PointNet++ [20], and adopt its MLP settings. We use leaky ReLU [15] as activation function. Cross-entropy loss is used as loss function with label smoothing. During training we augment the point clouds with random scaling in the range $[0.66, 1.5]$ and random translation in the range $[-0.2, 0.2]$. During testing, we report the best single model result without any voting operations.

Evaluation Metric. Same as previous work [19], we take overall classification accuracy as metric, formulated as $acc = N_{correct}/N_{all}$, where $N_{correct}$ denotes the correctly classified objects number, N_{all} denotes the total objects number.

Results. Similar to [3], we perform experiments under three settings: training and testing with vertical rotations (z/z), training with vertical rotations and testing with arbitrary rotations $(z/SO(3))$, and training and testing with arbitrary rotations $(SO(3)/SO(3))$. As shown in Table 1, the results are grouped, from top to bottom, by rotation-sensitive methods, rotation-robust (but not rotation-invariant) methods, LRF-based and PPF-based rotation-invariant methods. Though our method is not the best when compared to rotation-robust method (SFCNN [22]) in (z/z) and $(SO(3)/SO(3))$ cases, but maintain the same performance in all cases. And our method outperforms both LRF- and PPF-based rotation-invariant methods significantly, especially the PPF-based methods that are also based on angles and distances.

However, there still exists performance gap in (z/z) setting when compared to standard (rotation-sensitive) networks [19, 20, 25]. We speculate that the reason for this phenomenon is that these standard networks use some unique information under (z/z) setting, that is, overfitting this kind of scene.

4.2. Part Segmentation

Dataset. Part segmentation is a challenging task for fine-grained shape analysis. We evaluate our method for this task on ShapeNet part benchmark [29], which contains 16,881 shapes from 16 categories, annotated with 50 parts in total. We follow the data split in [19], and randomly pick 2048 points with only coordinates xyz as the input.

Implementation Details. We keep the same model architecture as PointNet++ [20] and concatenate the one-hot encoding of the object label to the last feature layer. We use ReLU [17] as activation function. Negative log likelihood loss is minimized during training. Other training and testing

Method	Inputs	Input size	z/z	z/SO(3)	SO(3)/SO(3)
PointNet [19]	pc	2048×3	79.3	43.0	73.9
PointNet++ [20]	pc	2048×3	**80.6**	45.9	75.5
DGCNN [25]	pc	2048×3	79.2	46.1	71.8
L2G-GCN [11]	pc	2048×3	-	77.2	77.3
RICNN [32]	pc	2048×3	-	75.3	75.5
SRINet [23]	pc	2048×3	77	77.0	77.0
Ours	pc	2048×3	79.3	**79.3**	**79.3**

Table 2. Part segmentation results on ShapeNet part dataset. We report the mIoU over all classes in three different train/test settings: (z/z), $(z/SO(3))$, and $(SO(3)/SO(3))$. The results are grouped, from top to bottom, by rotation-sensitive methods, LRF-based and PPF-based rotation-invariant methods. Our method outperforms other rotation-invariant methods by a large margin in all cases.

settings are kept same as in shape classification task.

Evaluation Metric. We use mean intersection-over-union (mIoU) over all classes as the evaluation metric, same as PointNet [19]. The mIoU can be formulated as:

$$mIoU = \frac{1}{C}\sum_{c=1}^{C}[\frac{1}{N_s^c}\sum_{s=1}^{N_s^c}(\frac{1}{N_p^s}\sum_{i=1}^{N_p^s}\frac{TP_i^s}{TP_i^s + FP_i^s + FN_i^s})] \tag{17}$$

where TP_i^s, FP_i^s, FN_i^s denote true positive, false positive, and false negative predictions for i-th part in shape s, N_p^s is the total part number in shape s, N_s^c is the total shape number in category c, and C is the number of categories. Note that, if the union of groundtruth and prediction points is empty, then count part IoU as 1.

Results. As we can see in Table 2, rotation-sensitive methods [19, 20, 25] drop sharply in $z/SO(3)$ case, and our method outperforms previous rotaion-invariant methods significantly in all cases. This result aligns well with the performance reported in the object classification task. Visualization of our prediction and the ground truth object parts are shown in Figure 4. We can see that the network produces same segmentation results with arbitrary rotations on the input point clouds. In Figure 5, we visualize the feature of a desk under two rotations for our method and AECNN [31]. When rotated around point O_1 which is selected by AECNN as reference point, the desk feature keeps the same for both methods. However, when rotated around another point O_2, the feature of AECNN changes distinctly, as shown in Figure 5(b), because the LRF changes when rotates around other points instead of the reference point. Therefore, this confirms our argument that AECNN is not a local rotation-invariant network.

4.3. Discussion

Effectiveness of SGM. The SGM descriptor, in essence, is the dense connected angle and distance information among points. We conduct an ablation experiment to explore the effectiveness of the SGM. As shown in Table 3, we can discover that the SGM ouperforms simplified PPFs by a large margin, showcasing that the SGM can indeed boost the performance of rotation-invariant network.

Features	Accurancy
distance	80.6
angle	85.2
distance+angle	86.6
SGM	90.0

Table 3. Illustration the effectiveness of SGM descriptor. Overall classification accuracy(%) on ModelNet40 [27] is reported.

Method	#params(M)	z/SO(3)
PointNet++ [20]	1.74	16.0
Spherical CNN [8]	0.50	78.6
RICNN [32]	0.70	86.4
AECNN(w/o align) [31]	1.94	89.6
SGMNet(*big*)	1.81	**90.0**
SGMNet(*middle*)	0.67	89.6
SGMNet(*small*)	0.44	89.3
SGMNet(*mini*)	0.23	88.8

Table 4. Comparison of model complexity. Overall classification accuracy(%) on ModelNet40 [27] under $z/SO(3)$ setting is reported. Our SGMNet is based on PointNet++ [20], and continuously cut the channel to reduce model complexity.

Model Complexity. For networks that rely on data augmentation to handle rotation issue, it requires more parameters to fit the enlarged input space. Models designed to be rotation-invariant should have lower complexity, especially for those whose local features are also rotation-invariant.

We conduct several experiments to demonstrate the above argument, the results are shown in Table 4. We illustrate it in two perspectives: 1). our proposed SGM is an compact and powerful local feature descriptor, since the performance only drops a little(0.4% to 1.2%) when continuously cut the model channel to reduce complexity, and SGMNet outperforms other rotation-invariant methods even with half the parameters; 2). the designed SGM feature descriptor with invariance property to local rotations leads to lower model complexity than those only invariant to

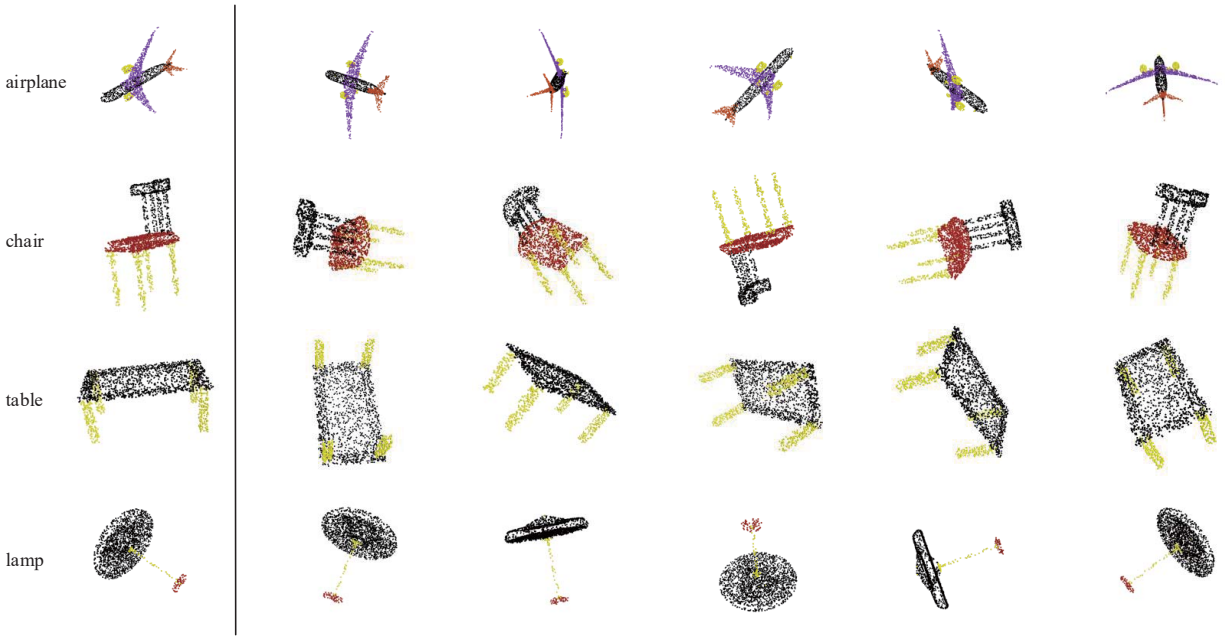


Figure 4. Qualitative results of the proposed method on ShapeNet dataset. From top to bottom, we show part segmentation results for airplane, chair, table and lamp class. From left to right are ground truth and segmentation results with the input point clouds arbitrarily rotated during testing. We can see that the network produces same segmentation results with arbitrary rotations on the input point clouds.

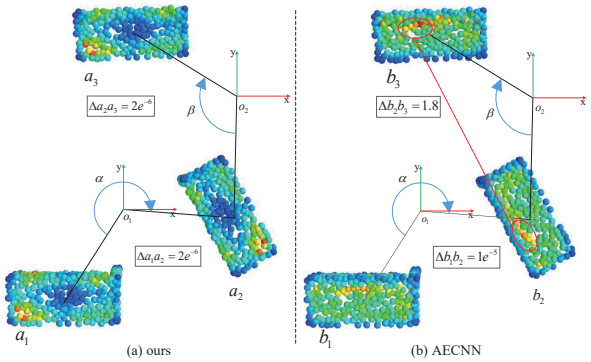


Figure 5. Illustration of local rotation-invariance(RI) property: a_1 and b_1 are the same desk. We first rotate a_1 and b_1 around origin point O_1 (selected by AECNN [31] as reference point) to get a_2 and b_2, then rotate a_2 and b_2 around another arbitrary point O_2 to get a_3 and b_3. Color indicates norm of features. For our method, differences between a_1 and a_2 ($\Delta a_1 a_2$), a_2 and a_3 ($\Delta a_2 a_3$) in feature space are all near 0, showing perfect local RI property. For AECNN [31], difference between b_1 and b_2 ($\Delta b_1 b_2$) is still near 0 while b_2 and b_3 ($\Delta b_2 b_3$) is not. Pay attention to the clear distinctions between 2 red ellipses in b_2 and b_3.

global rotations [31], and thus has wider application potential especially under computationally resource-constrained circumstances.

Generalizability of SGM. In Table 5, we present the results of SGM integrated as a drop-in module into some popular point cloud analysis networks [20, 14, 25]. All three networks achieve decent performance on ModelNet40 [27] and maintain the same performance under (z/z), $(z/SO(3))$, and $(SO(3)/SO(3))$ settings, showing the generalizability of the SGM.

Method	z/z	z/SO(3)	SO(3)/SO(3)
PointNet++ [20] + SGM	90.0	90.0	90.0
RSCNN [14] + SGM	88.5	88.5	88.5
DGCNN [25] + SGM	88.9	88.9	88.9

Table 5. Illustration on the generalizability of SGM descriptor when integrated into three different point cloud analysis networks. Overall classification accuracy(%) on ModelNet40 [27] is reported.

5. Conclusion

We propose a simple but effective rotation-invariant descriptor, namely **s**orted **G**ram **m**atrix(SGM), for local point cloud representation. We densely connect every pair of points in a local neighborhood to obtain the full geometrical description through a Gram matrix, and mathematically prove that it is rotation-invariant and sufficient to model the inherent structure of the point cloud patch. Extensive experiments performed on classification and segmentation tasks with qualitative and quantitative results demonstrate the effectiveness, efficiency and generalizability of our SGM descriptor.

References

[1] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018. 1

[2] Tolga Birdal and Slobodan Ilic. Point pair features based object detection and pose estimation revisited. In *2015 International Conference on 3D Vision*, pages 527–535. IEEE, 2015. 3, 4

[3] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4994–5002, 2019. 2, 3, 4, 6

[4] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018. 3

[5] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. pages 620–638, 09 2018. 3

[6] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 195–205, 2018. 4

[7] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010. 2, 3

[8] 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. 3, 6, 7

[9] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. In *European conference on computer vision*, pages 834–848. Springer, 2016. 3

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015. 2

[11] Seohyun Kim, Jaeyoo Park, and Bohyung Han. Rotation-invariant local-to-global representation learning for 3d point cloud. *arXiv preprint arXiv:2010.03318*, 2020. 2, 3, 6, 7

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

[13] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on χ-transformed points. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 828–838, 2018. 1, 3

[14] Yongcheng Liu, Bin Fan, S. Xiang, and C. Pan. Relation-shape convolutional neural network for point cloud analysis. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8887–8896, 2019. 8

[15] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013. 6

[16] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. 2

[17] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010. 6

[18] Thomas Pumir, Amit Singer, and Nicolas Boumal. The generalized orthogonal procrustes problem in the high noise regime. *arXiv preprint arXiv:1907.01145*, 2019. 4

[19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 1, 2, 5, 6, 7

[20] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 1, 3, 5, 6, 7, 8

[21] Yongming Rao, Jiwen Lu, and J. Zhou. Spherical fractal convolutional neural networks for point cloud recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 452–460, 2019. 3

[22] Yongming Rao, Jiwen Lu, and Jie Zhou. Spherical fractal convolutional neural networks for point cloud recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 452–460, 2019. 6

[23] Xiao Sun, Zhouhui Lian, and Jianguo Xiao. Srinet: Learning strictly rotation-invariant representations for point cloud classification and segmentation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 980–988, 2019. 2, 3, 6, 7

[24] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. 1, 3

[25] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1, 2, 3, 6, 7, 8

[26] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 1, 3

[27] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2, 6, 7, 8

[28] Zelin Xiao, Hongxin Lin, Renjie Li, Lishuai Geng, Hongyang Chao, and Shengyong Ding. Endowing deep 3d models with rotation invariance based on principal component analysis. In *2020 IEEE International Conference on*

Multimedia and Expo (ICME), pages 1–6. IEEE, 2020. 2, 3, 6

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

[30] Yang You, Yujing Lou, Qi Liu, Yu-Wing Tai, Lizhuang Ma, Cewu Lu, and Wei-Ming Wang. Pointwise rotation-invariant network with adaptive sampling and 3d spherical voxel convolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:12717–12724, 04 2020. 3

[31] Junming Zhang, Ming-Yuan Yu, Ram Vasudevan, and Matthew Johnson-Roberson. Learning rotation-invariant representations of point clouds using aligned edge convolutional neural networks. In *2020 International Conference on 3D Vision (3DV)*, pages 200–209. IEEE, 2020. 2, 3, 7, 8

[32] Zhiyuan Zhang, Binh-Son Hua, David W Rosen, and Sai-Kit Yeung. Rotation invariant convolutions for 3d point clouds deep learning. In *2019 International Conference on 3D Vision (3DV)*, pages 204–213. IEEE, 2019. 2, 3, 4, 6, 7

[33] Yongheng Zhao, Tolga Birdal, Jan Eric Lenssen, Emanuele Menegatti, Leonidas Guibas, and Federico Tombari. Quaternion equivariant capsule networks for 3d point clouds. In *European Conference on Computer Vision*, pages 1–19. Springer, 2020. 2, 3