# MeshNet: Mesh Neural Network for 3D Shape Representation

PinHan Chen

# Abstract

- Mesh is an important and powerful type of 3D shapes
- The complexity and irregularity of mesh data
- MeshNet is proposed to learn 3D shape representation from mesh data

# Introduction

Several types of 3D data:

- Volumetric grid
- Multi-view
- Point cloud
- **Mesh**

# Introduction(cont'd)

Key contributions of this paper:

- A neural network using mesh for 3D shape representation and design blocks for capturing and aggregating features of polygon faces in 3D shapes
- Expensive experiments to evaluate the performance of the proposed method, and the experimental results show that the proposed method performs well on the 3D shape classification and retrieval task

# Related Work

## Mesh Feature Extraction

- A symbolic method for calculating the integral properties of arbitrary nonconvex polyhedra(1984)
- Efficient Feature Extraction for 2D/3D Objects in Mesh Representation(2001)
- Multiresolution Feature Extraction for Unstructured Meshes(2001)
- Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors(2003)
- Surface Feature Detection and Description with Applications to Mesh Matching(2009)
- Intrinsic Shape Context Descriptors for Deformable Shapes(2012)

# Related Work

## Deep Learning Methods for 3D Shape Representation

Voxel-based methods:

- 3d ShapeNets: A Deep Representation for Volumetric Shapes(2015)
- FPNN: Field Probing Neural Networks for 3D Data(2016)
- Voting for Voting in Online Point Cloud Object Detection(2015)
- O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis(2017)

# Related Work

## Deep Learning Methods for 3D Shape Representation

View-based methods:

- Multi-View Convolutional Neural Networks for 3D Shape Recognition(2015)
- Group-View Convolutional Neural Networks for 3D Shape Recognition(2018)

# Related Work

## Deep Learning Methods for 3D Shape Representation

Point-based methods:

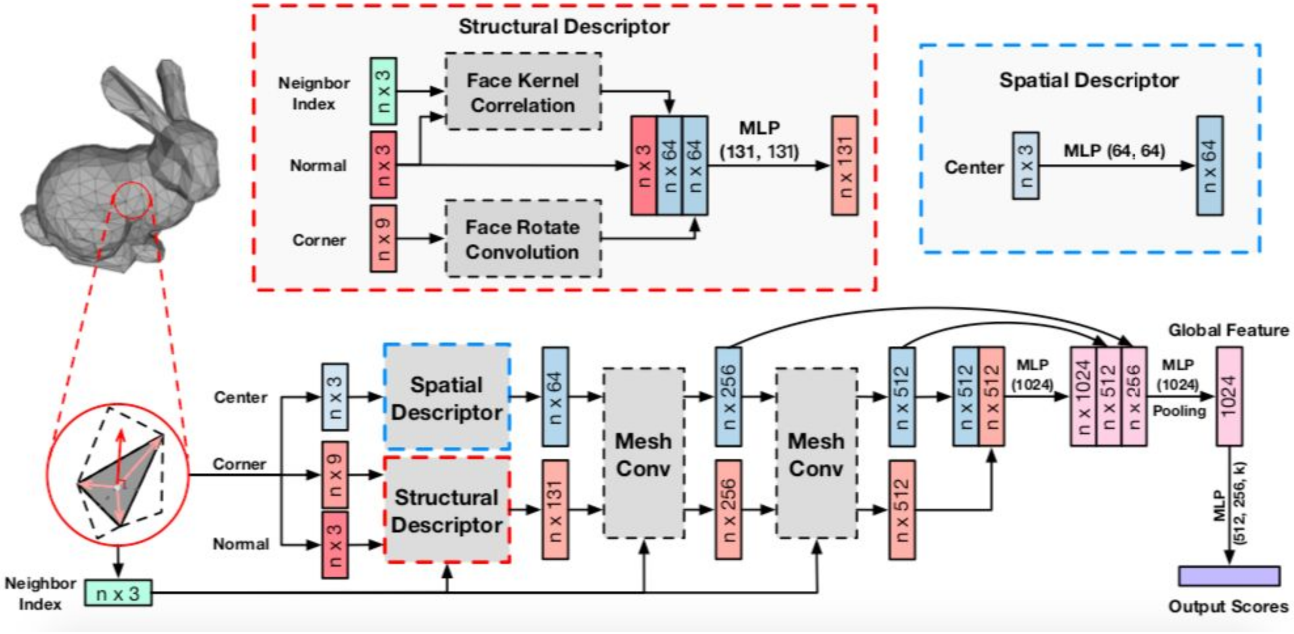
- PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation(2017)
- PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space(2017)
- SO-Net: Self-Organizing Network for Point Cloud Analysis(2018)
- Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling(2017)
- PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation(2018)
- Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models(2017)

# Related Work
## Deep Learning Methods for 3D Shape Representation

Fusion methods:

- Fusionnet: 3D Object Classification Using Multiple Data Representations(2016)
- PVnet: A Joint Convolutional Network of Point Cloud and Multi-view for 3D Shape Recognition(2018)

# Overall Design of MeshNet



# Overall Design of MeshNet(cont’d)

- Regard face as the unit
- Split feature of face

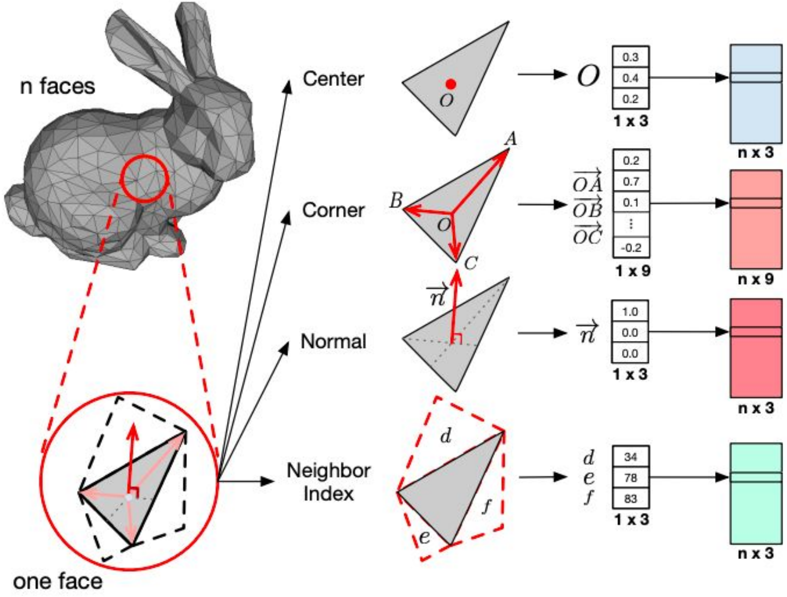
# Mesh Information

Face Information

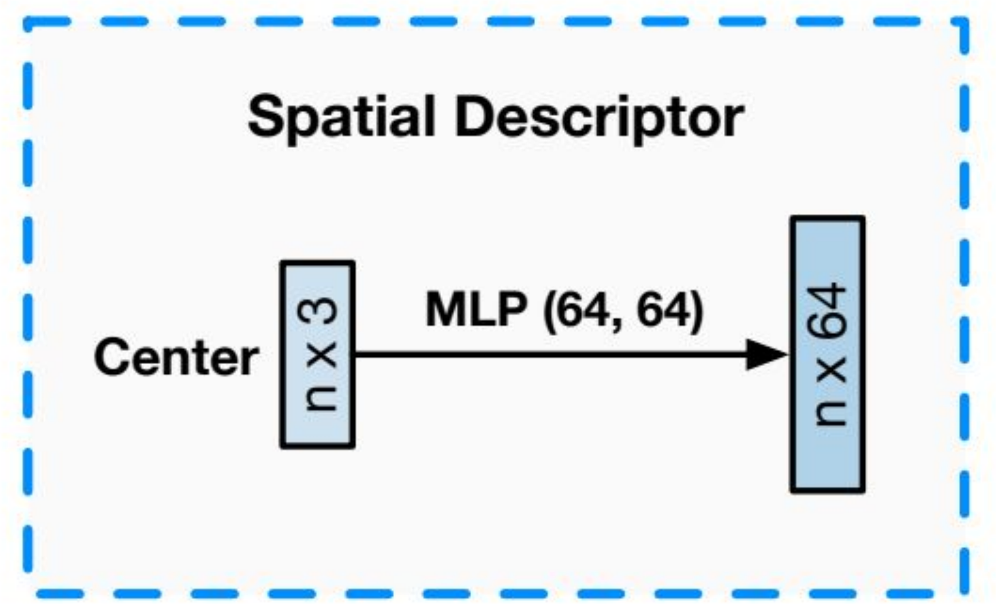
- Center
- Corner
- Normal (unit)

Neighbor Information

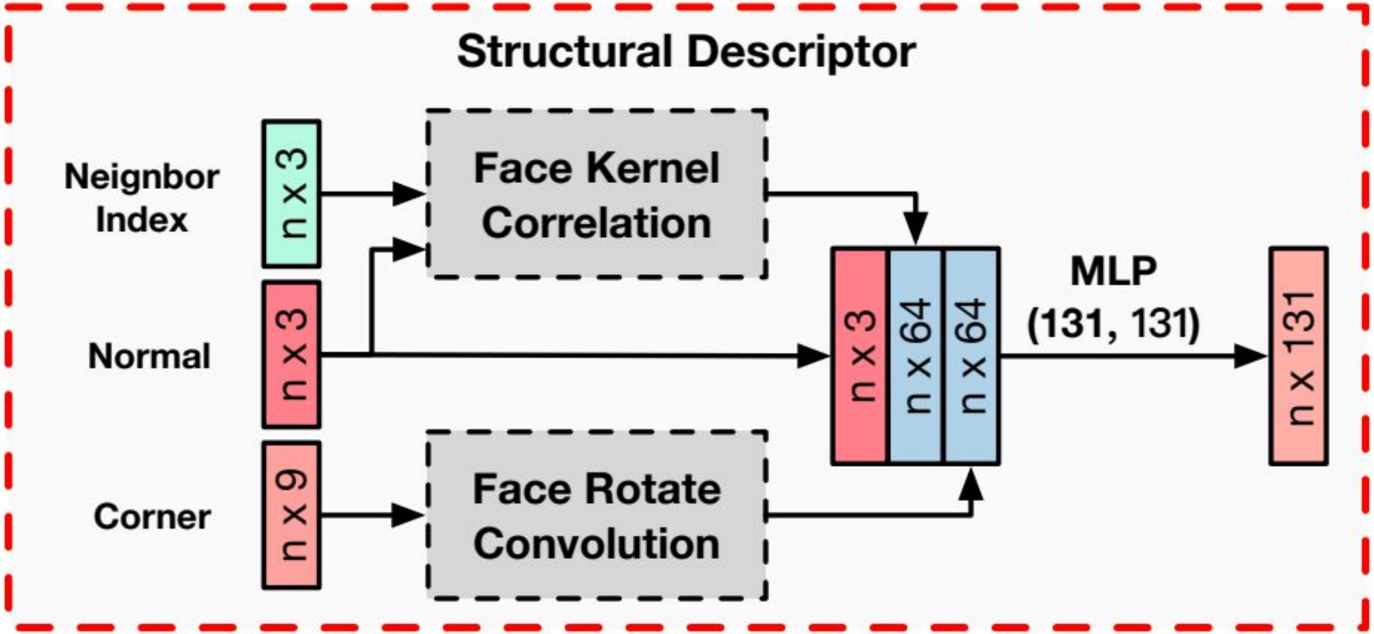
- Neighbor index



# Spatial Descriptor



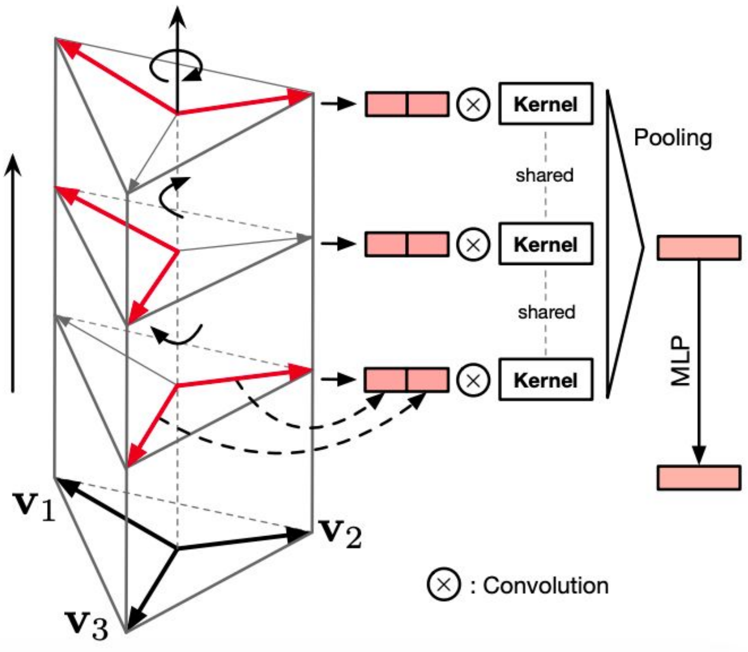
# Structural Descriptor



# Face Rotate Convolution

$$g(\frac{1}{3}(f(\mathbf{v}_1,\mathbf{v}_2)+f(\mathbf{v}_2,\mathbf{v}_3)+f(\mathbf{v}_3,\mathbf{v}_1)))$$

$$f(\cdot,\cdot):\mathbb{R}^3\times\mathbb{R}^3\rightarrow\mathbb{R}^{K_1} \text{ and } g(\cdot):\mathbb{R}^{K_1}\rightarrow\mathbb{R}^{K_2}$$



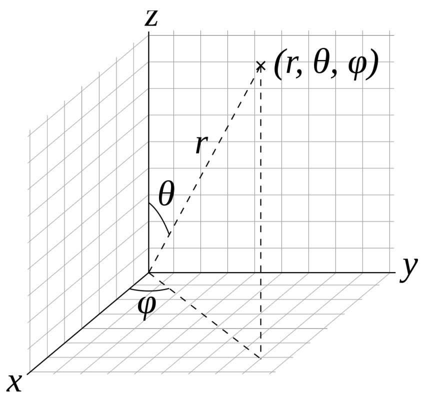
# Face Rotate Convolution(cont’d)

```
class FaceRotateConvolution(nn.Module):

    def __init__(self):
        super(FaceRotateConvolution, self).__init__()
        self.rotate_mlp = nn.Sequential(
            nn.Conv1d(6, 32, 1),
            nn.BatchNorm1d(32),
            nn.ReLU(),
            nn.Conv1d(32, 32, 1),
            nn.BatchNorm1d(32),
            nn.ReLU()
        )
        self.fusion_mlp = nn.Sequential(
            nn.Conv1d(32, 64, 1),
            nn.BatchNorm1d(64),
            nn.ReLU(),
            nn.Conv1d(64, 64, 1),
            nn.BatchNorm1d(64),
            nn.ReLU()
        )

    def forward(self, corners):

        fea = (self.rotate_mlp(corners[:, :6]) +
               self.rotate_mlp(corners[:, 3:9]) +
               self.rotate_mlp(torch.cat([corners[:, 6:], corners[:, :3]], 1))) / 3

        return self.fusion_mlp(fea)
```

# Face Kernel Correlation

- Capturing the “outer” structure of faces and explore the environments where faces locate
- Model vectors of kernels with parameters in the spherical coordinate system

$$\begin{cases} x = \sin\theta\cos\phi \\ y = \sin\theta\sin\phi \\ z = \cos\theta \end{cases}$$



# Face Kernel Correlation(cont'd)

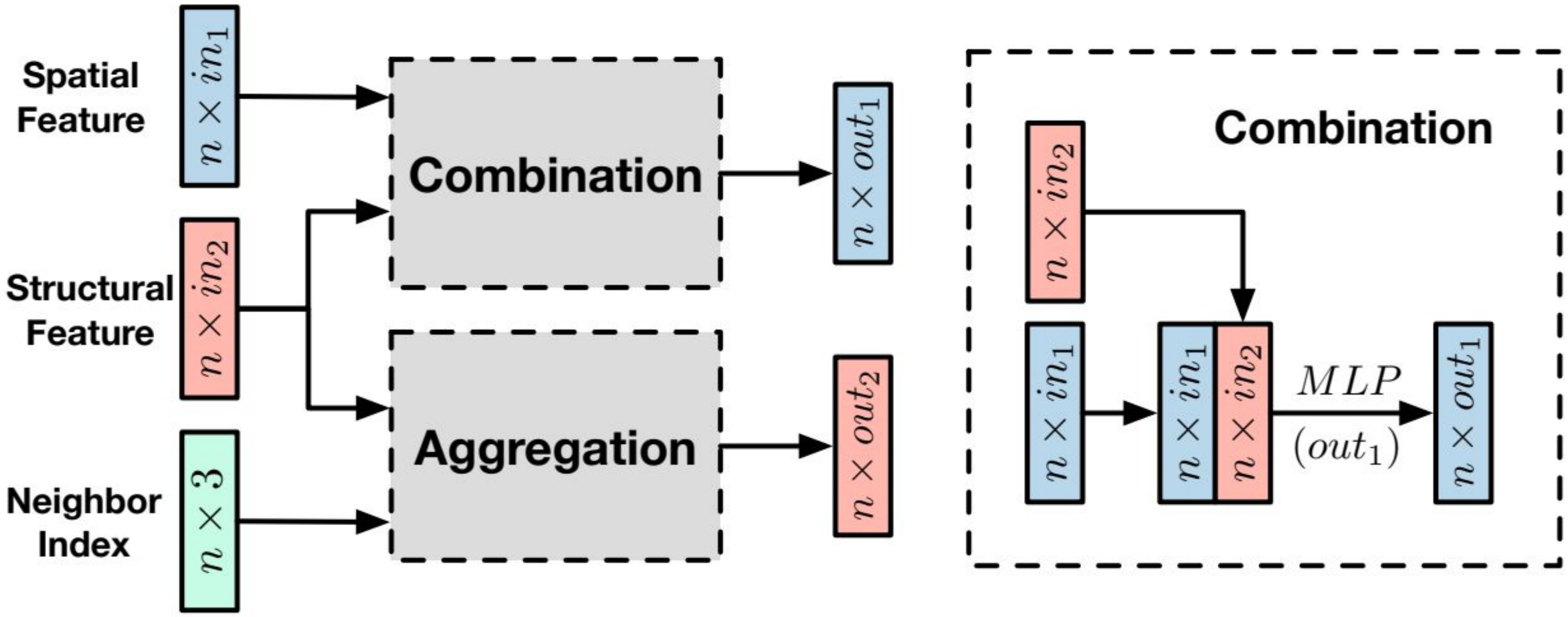
Define the kernel correlation between the i-th face and the k-th kernel as follows:

$$KC(i,k) = \frac{1}{|\mathcal{N}_i||\mathcal{M}_k|} \sum_{\mathbf{n}\in\mathcal{N}_i} \sum_{\mathbf{m}\in\mathcal{M}_k} K_\sigma(\mathbf{n},\mathbf{m})$$

$$K_\sigma(\mathbf{n},\mathbf{m}) = \exp\left(-\frac{\|\mathbf{n}-\mathbf{m}\|^2}{2\sigma^2}\right)$$

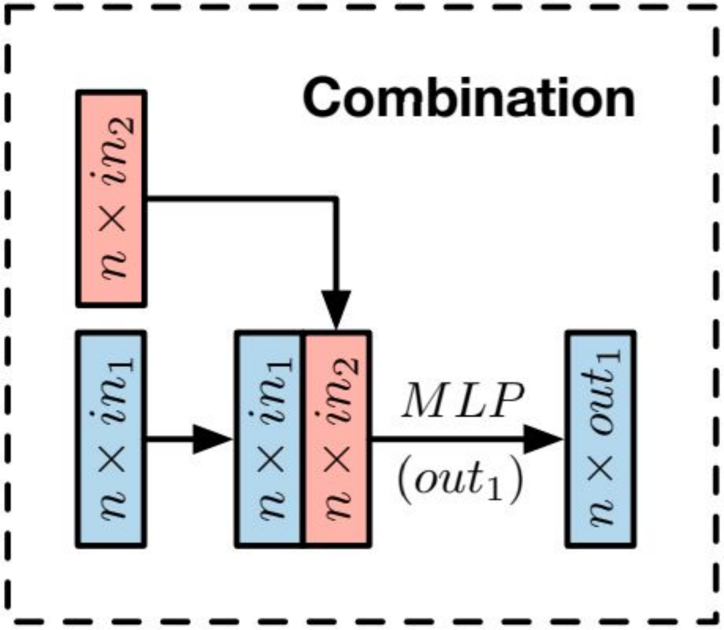
More similar pairs will get higher values.

# Mesh Convolution

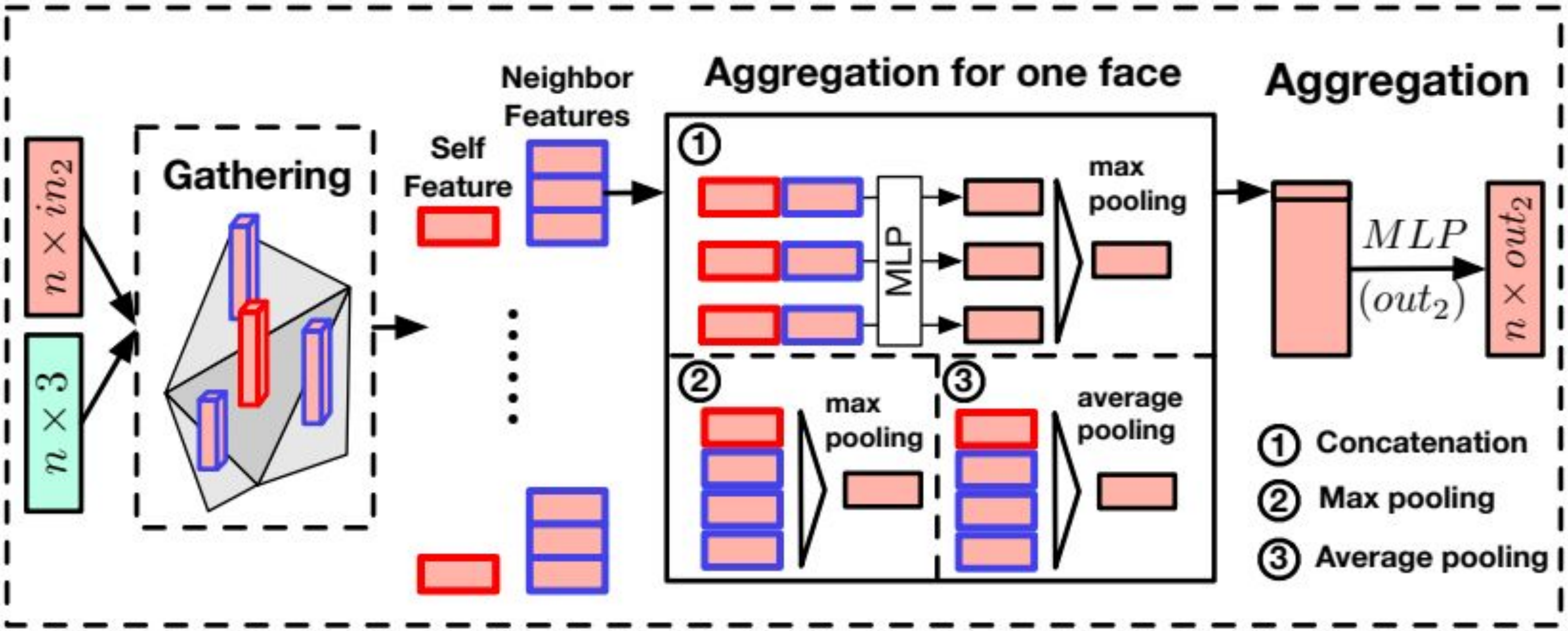


# Combination of Spatial and Structural Features

The combination result contains both spatial and structural information.

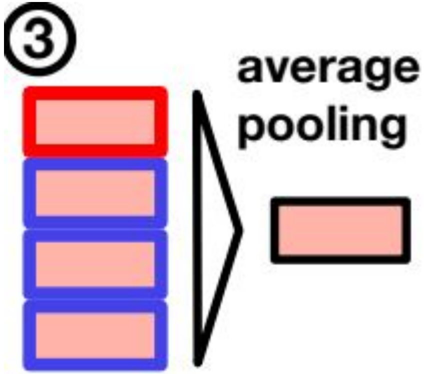


# Aggregation of Structural Feature



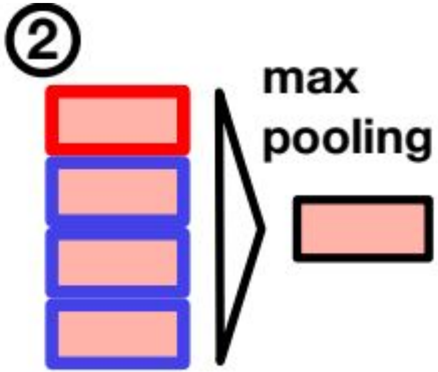
# Aggregation methods

## Average pooling



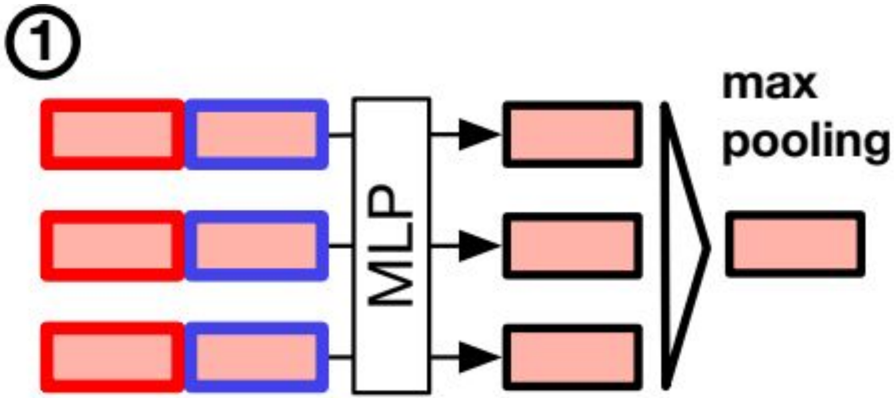
# Aggregation methods

## Max pooling



# Aggregation methods

## Concatenation



# Implementation Details

- The spatial descriptor contains fully-connected layers (64, 64) and output a initial spatial feature with length of 64.
- In the face rotate convolution, they set K1 = 32 and K2 = 64, and correspondingly, the functions f(., .) and g(.) are implemented by fully-connected layers (32, 32) and (64, 64).
- In the face kernel correlation, they set M = 64(64 kernels) and $\sigma$ = 0.2.

# Experiments

## 3D shape classification and retrieval

| Method | Modality | Acc (%) | mAP (%) |
|---|---|---|---|
| 3DShapeNets (Wu et al. 2015) | volume | 77.3 | 49.2 |
| VoxNet (Maturana and Scherer 2015) | volume | 83.0 | - |
| FPNN (Li et al. 2016) | volume | 88.4 | - |
| LFD (Chen et al. 2003) | view | 75.5 | 40.9 |
| MVCNN (Su et al. 2015) | view | 90.1 | 79.5 |
| Pairwise (Johns, Leutenegger, and Davison 2016) | view | 90.7 | - |
| PointNet (Qi et al. 2017a) | point | 89.2 | - |
| PointNet++ (Qi et al. 2017b) | point | 90.7 | - |
| Kd-Net (Klokov and Lempitsky 2017) | point | 91.8 | - |
| KCNet (Shen et al. ) | point | 91.0 | - |
| SPH (Kazhdan, Funkhouser, and Rusinkiewicz 2003) | mesh | 68.2 | 33.3 |
| MeshNet | mesh | 91.9 | 81.9 |

Table 1: Classification and retrieval results on ModelNet40.

| | | | | | | |
|---|---|---|---|---|---|---|
| Spatial | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Structural-FRC | ✓ | | | ✓ | ✓ | ✓ |
| Structural-FKC | ✓ | | ✓ | | ✓ | ✓ |
| Mesh Conv | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Accuracy (%) | 83.5 | 88.2 | 87.0 | 89.9 | 90.4 | 91.9 |

Table 2: Classification results of ablation experiments on ModelNet40.

| Aggregation Method | Accuracy (%) |
|---|---|
| Average Pooling | 90.7 |
| Max Pooling | 91.1 |
| Concatenation | 91.9 |

Table 3: Classification results of different aggregation methods on ModelNet40.

# Experiments

## On the Number of Faces

| Number of Faces | Proportion (%) | Accuracy (%) |
|---|---|---|
| $[1000, 1024)$ | 69.48 | 92.00 |
| $[800, 1000)$ | 6.90 | 92.35 |
| $[600, 800)$ | 4.70 | 93.10 |
| $[400, 600)$ | 6.90 | 91.76 |
| $[200, 400)$ | 6.17 | 90.79 |
| $[0, 200)$ | 5.84 | 90.97 |

Table 4: Classification results of groups with different number of faces on ModelNet40.

# Experiments

## On the Time and Space Complexity

| Method | #params (M) | FLOPs / sample (M) |
|---|---|---|
| PointNet (Qi et al. 2017a) | 3.5 | 440 |
| Subvolume (Qi et al. 2016) | 16.6 | 3633 |
| MVCNN (Su et al. 2015) | 60.0 | 62057 |
| MeshNet | 4.25 | 509 |

Table 5: Time and space complexity for classification.

# Experiments

## Feature Visualization

Figure 6: **Feature visualization of structural feature.** Models from the same column are colored with their values of the same channel in features. **Left**: Features from the face rotate convolution. **Right**: Features from the face kernel correlation.

# Conclusions

In this paper, the proposed mesh neural network learns on mesh data directly for 3D shape representation. It is also able to solve the complexity and irregularity problem of mesh data and conduct 3D shape representation well.