PointConv: Deep Convolutional Networks on 3D Point Clouds

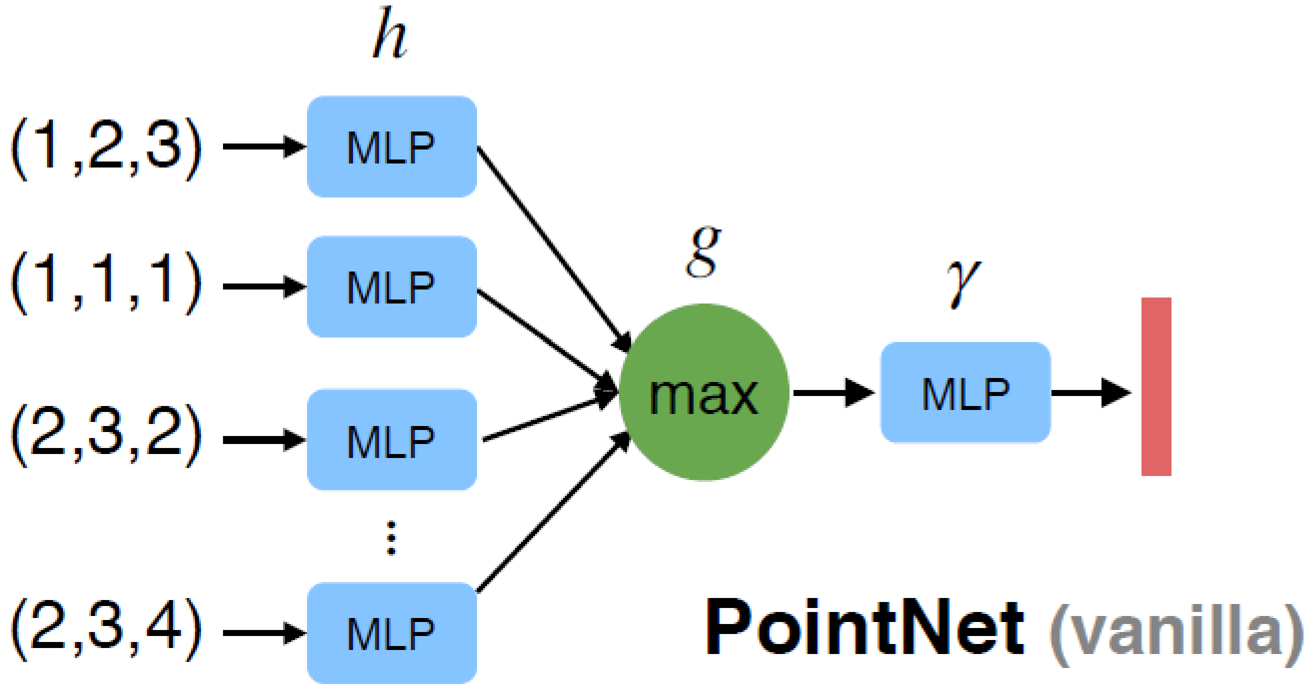
Wenxuan Wu, Zhongang Qi, Li Fuxin
CORIS Institute, Oregon State University
wuwen, qiz, lif@oregonstate.edu

3D 点云处理的理想性质

1. computationally efficient
2. invariant to the order of points in the point cloud
3. robust to different samplings and varying densities
4. translation invariant
5. …

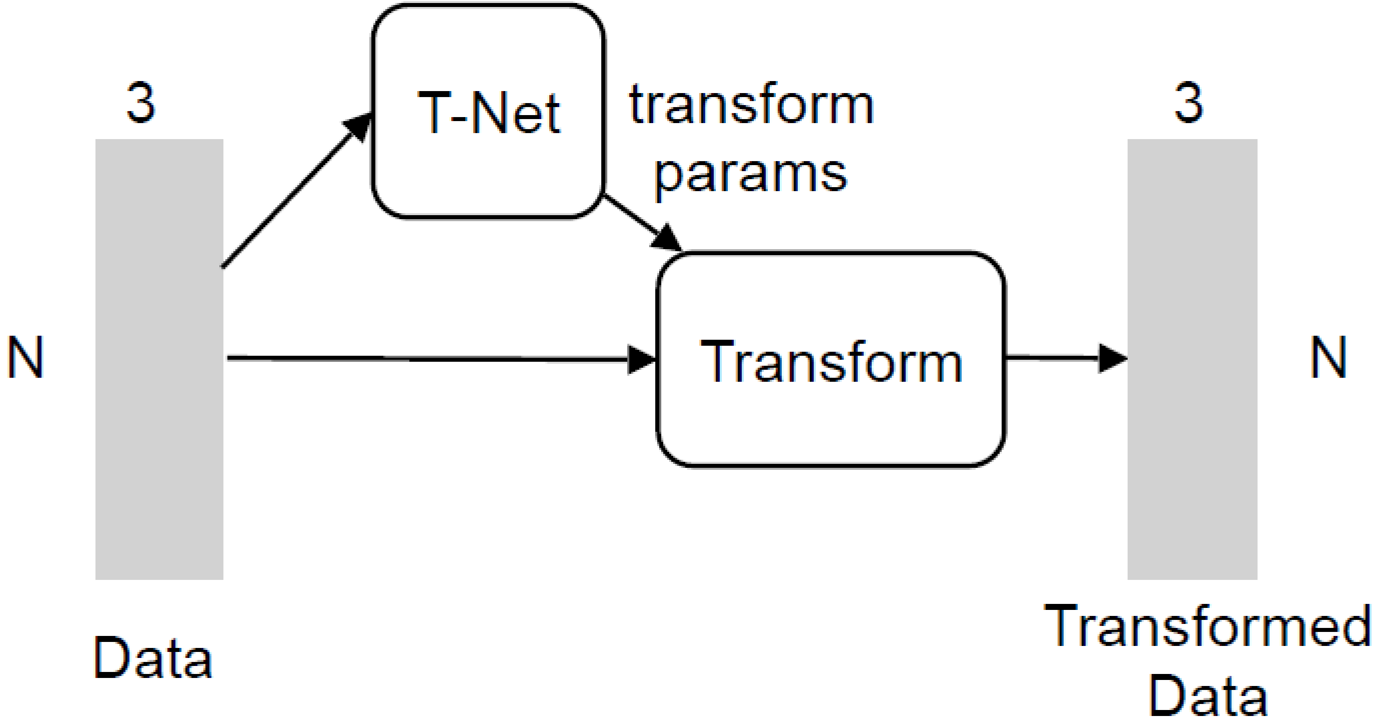
PointNet & PointNet++

- Basic Architecture



PointNet & PointNet++

- Input Alignment by Transformer Network

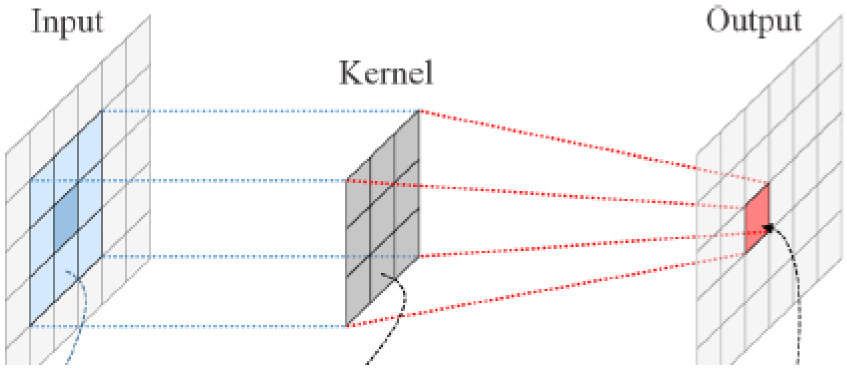


3D 点云处理的理想性质

1. computationally efficient √
2. invariant to the order of points in the point cloud √
3. robust to different samplings and varying densities √
4. translation invariant √
5. …

但不是卷积！！

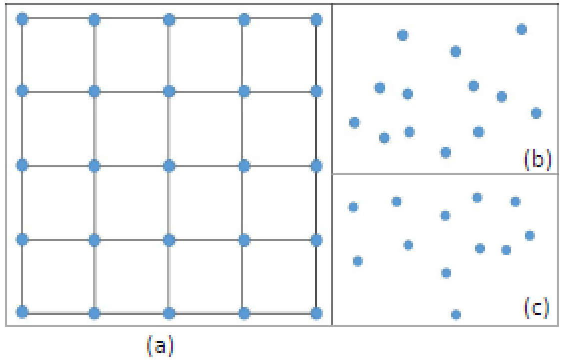
2D图像卷积



$$W \in R^{C_{out} \times k \times k \times C_{in}}$$

$$F_{c_{out}} = \sum_{c_{in}} \sum_{i,j} W_{i,j}^{c_{out},c_{in}} F_{i,j}^{c_{in}}$$

$p(i,j)$



$$F_{c_{out}} = \sum_{c_{in}} \sum_{i,j} p_{c_{out}}^{c_{in}}(i,j) F_{i,j}^{c_{in}}$$

位置函数 特征(函数)

3D PointConv

- 连续

$$Conv(W,F)_{xyz} = \iiint_{(\delta_x,\delta_y,\delta_z)\in G} W(\delta_x,\delta_y,\delta_z)F(x+\delta_x,y+\delta_y,z+\delta_z)d\delta_x\delta_y\delta_z$$

- 离散

$$PointConv(S,W,F)_{xyz} = \sum_{(\delta_x,\delta_y,\delta_z)\in G} S(\delta_x,\delta_y,\delta_z)W(\delta_x,\delta_y,\delta_z)F(x+\delta_x,y+\delta_y,z+\delta_z)$$

3D PointConv

- 离散

$$PointConv(S, W, F)_{xyz} = \sum_{(\delta_x, \delta_y, \delta_z) \in G} S(\delta_x, \delta_y, \delta_z) W(\delta_x, \delta_y, \delta_z) F(x + \delta_x, y + \delta_y, z + \delta_z)$$

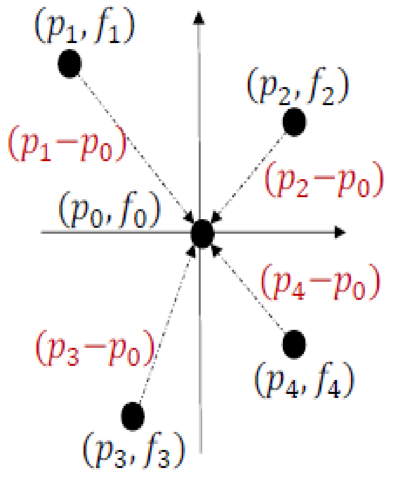
S 表示逆密度系数函数。连续函数 W 可以用多层感知器（MLP）近似。
函数 W 的输入是以 (x, y, z) 为中心的 3D 邻域内的 3D 点的相对坐标,
输出是每个点对应的特征 F 的权重

$$W \in R^{C_{out} \times k \times C_{in}}$$

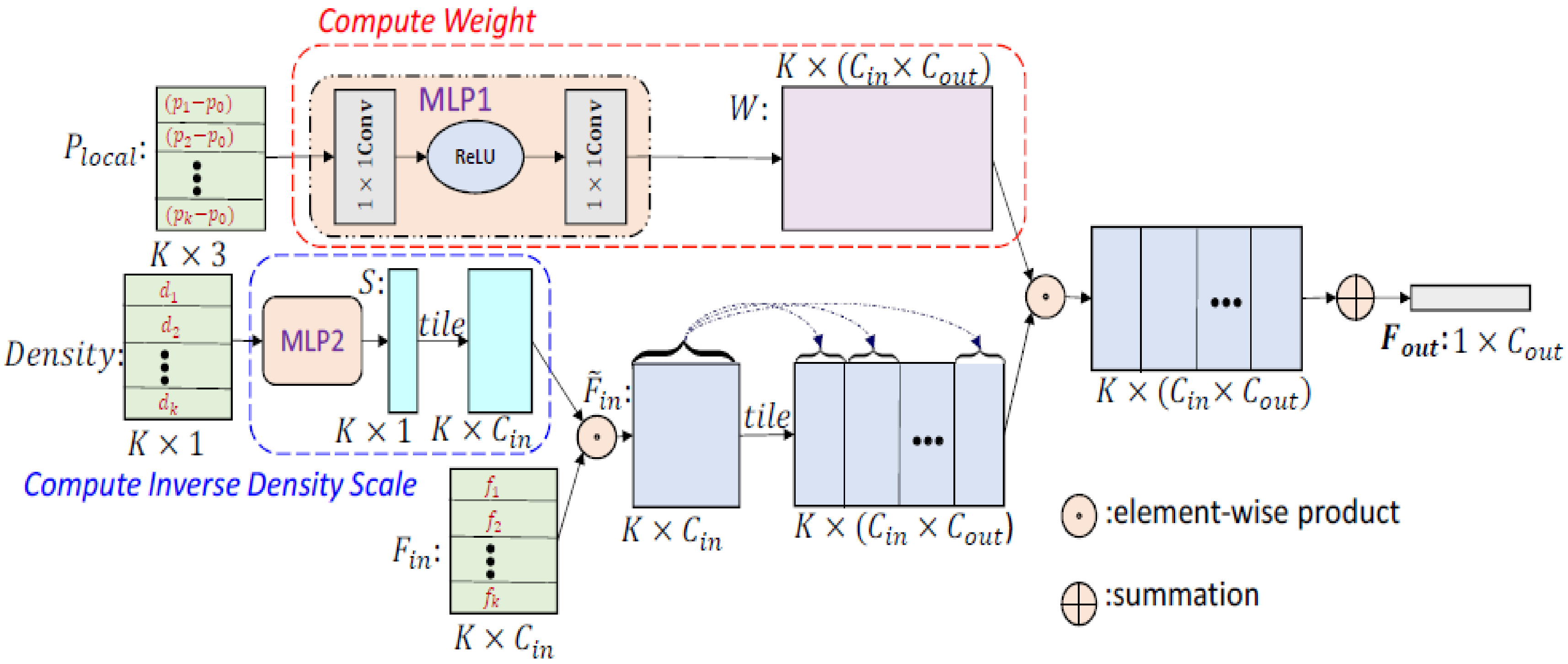
$$\mathbf{F}_{out} = \sum_{k=1}^{K} \sum_{c_{in}=1}^{C_{in}} S(k) \mathbf{W}(k, c_{in}) F_{in}(k, c_{in})$$

$$F_{c_{out}} = \sum_{c_{in}} \sum_{i,j} p_{c_{out}}^{c_{in}}(i, j) F_{i,j}^{c_{in}}$$

3D PointConv



(a)



(b)

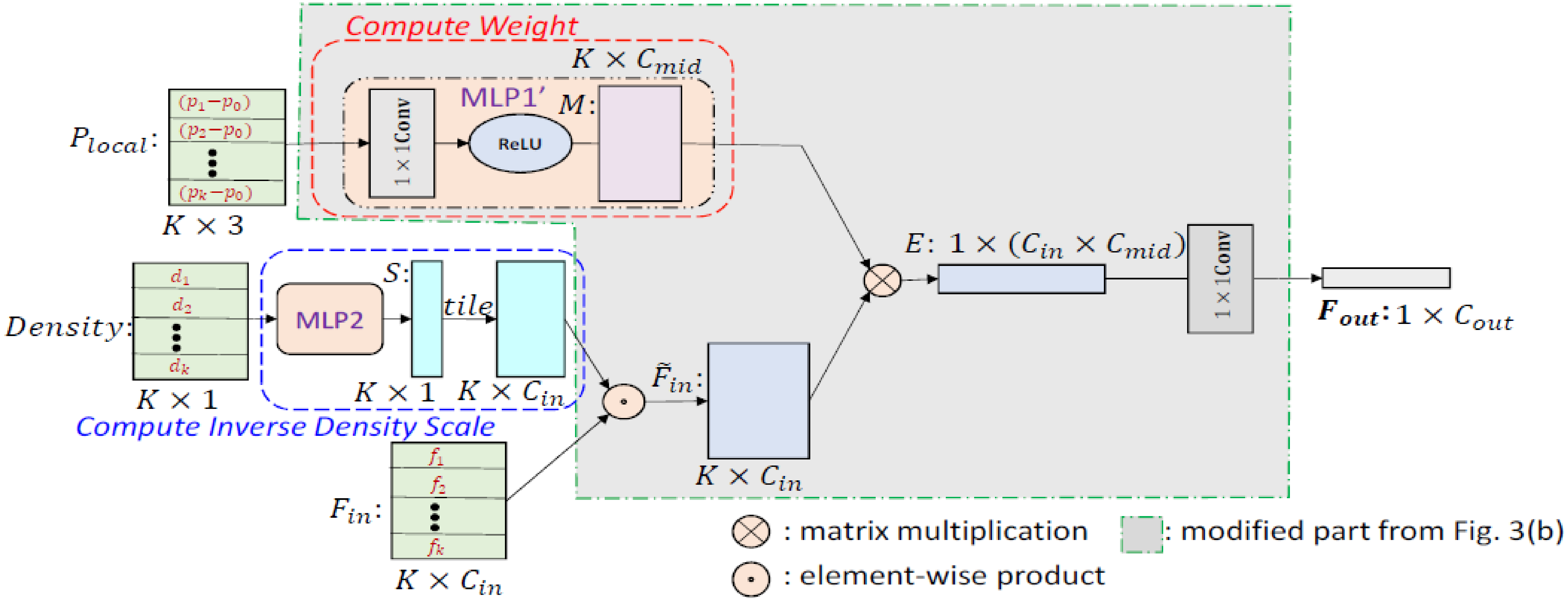
3D 点云处理的理想性质

1. computationally efficient ×
2. invariant to the order of points in the point cloud √
3. robust to different samplings and varying densities ×(scale)
4. translation invariant ×(rotate)
5. …

Efficient PointConv

- For a point cloud, each local region shares the same weight functions which can be learned using MLP. However, weights computed from the weight functions at different points are different. The size of the weights filters generated by the MLP is
  - $B \times N \times K \times (Cin \times Cout)$.
- Suppose B = 32, N = 512, K = 32, Cin = 64, Cout = 64, and the filters are stored with single point precision. Then, the memory size for the filters is 8GB for only one layer.

Lemma 1 *The PointConv is equivalent to the following formula:* $\mathbf{F}_{out} = Conv_{1\times1}(\mathbf{H}, (\mathbf{S}\cdot\mathbf{F}_{in})^T \otimes \mathbf{M})$ *where* $\mathbf{M} \in \mathbb{R}^{K\times C_{mid}}$ *is the input to the last layer in the MLP for computing the weight function, and* $\mathbf{H} \in \mathbb{R}^{C_{mid}\times(C_{in}\times C_{out})}$ *is the weights of the last layer in the same MLP,* $Conv_{1\times1}$ *is* 1×1 *convolution.*



Experiment

- ModelNet40
- ShapeNet
- ScanNet
- CIFAR-10

Experiment

Table 1. **ModelNet40 Classification Accuracy**

Method	Input	Accuracy(%)
Subvolume [27]	voxels	89.2
ECC [33]	graphs	87.4
Kd-Network [18]	1024 points	91.8
PointNet [26]	1024 points	89.2
PointNet++ [28]	1024 points	90.2
PointNet++ [28]	5000 points+normal	91.9
SpiderCNN [44]	1024 points+normal	92.4
PointConv	1024 points+normal	**92.5**

Table 2. **Results on ShapeNet part dataset.** Class avg. is the mean IoU averaged across all object categories, and inctance avg. is the mean IoU across all objects.

	class avg.	instance avg.
SSCNN [45]	82.0	84.7
Kd-net [18]	77.4	82.3
PointNet [26]	80.4	83.7
PointNet++[28]	81.9	85.1
SpiderCNN [44]	82.4	85.3
SPLATNet$_{3D}$ [35]	82.0	84.6
SSCN [7]	-	**86.0**
PointConv	**82.8**	85.7

Experiment

Table 3. **Semantic Scene Segmentation results on ScanNet**

Method	mIoU(%)
ScanNet [5]	30.6
PointNet++ [28]	33.9
SPLAT Net [35]	39.3
Tangent Convolutions [37]	43.8
PointConv	**55.6**

Table 4. **CIFAR-10 Classification Accuracy**

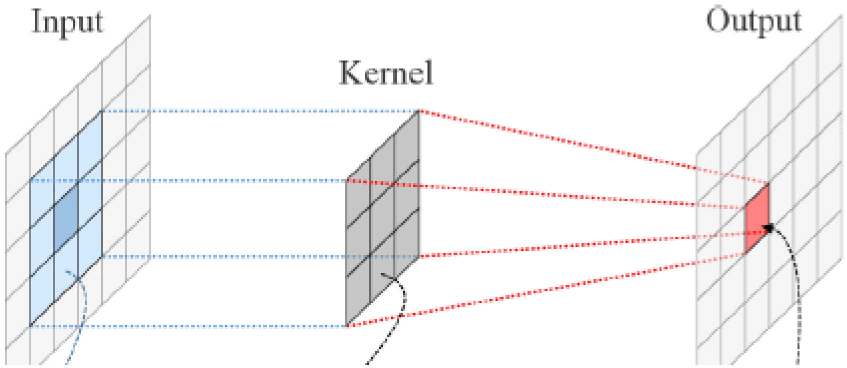
	Accuracy(%)
Image Convolution	88.52
AlexNet [20]	89.00
VGG19 [34]	93.60
PointCNN [21]	80.22
SpiderCNN [44]	77.97
PointConv(5-layer)	89.13
PointConv(VGG19)	93.19

KPConv: Flexible and Deformable Convolution for Point Clouds

Hugues Thomas[1] Charles R. Qi[2] Jean-Emmanuel Deschaud[1] Beatriz Marcotegui[1]
François Goulette[1] Leonidas J. Guibas[2,3]

[1]Mines ParisTech [2]Facebook AI Research [3]Stanford University

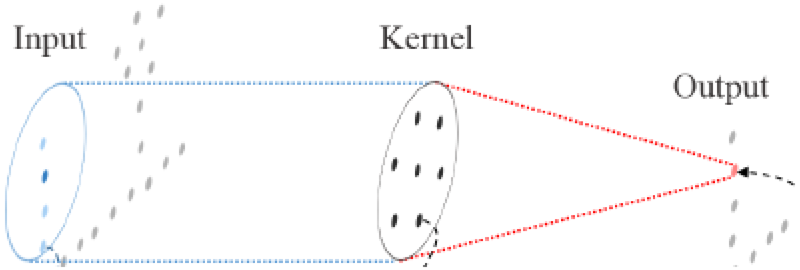
2D图像卷积



$$W \in R^{C_{out}\times K\times K\times C_{in}}$$

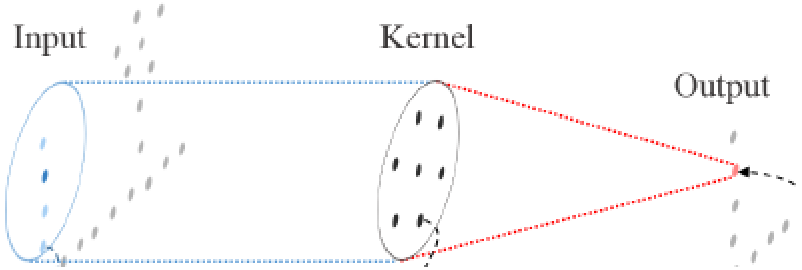
$$F_{c_{out}} = \sum_{c_{in}} \sum_{i,j} W_{i,j}^{c_{out},c_{in}} F_{i,j}^{c_{in}}$$

$$\sum_{k\in K} W_{k,k}\delta((k,k),(i,j))$$



$$F_{c_{out}} = \sum_{c_{in}} \sum_{x_i\in R_k} \sum_{k} W_k^{c_{in}} h(p_k, x_i) F_{x_i}^{c_{in}}$$

2D图像卷积



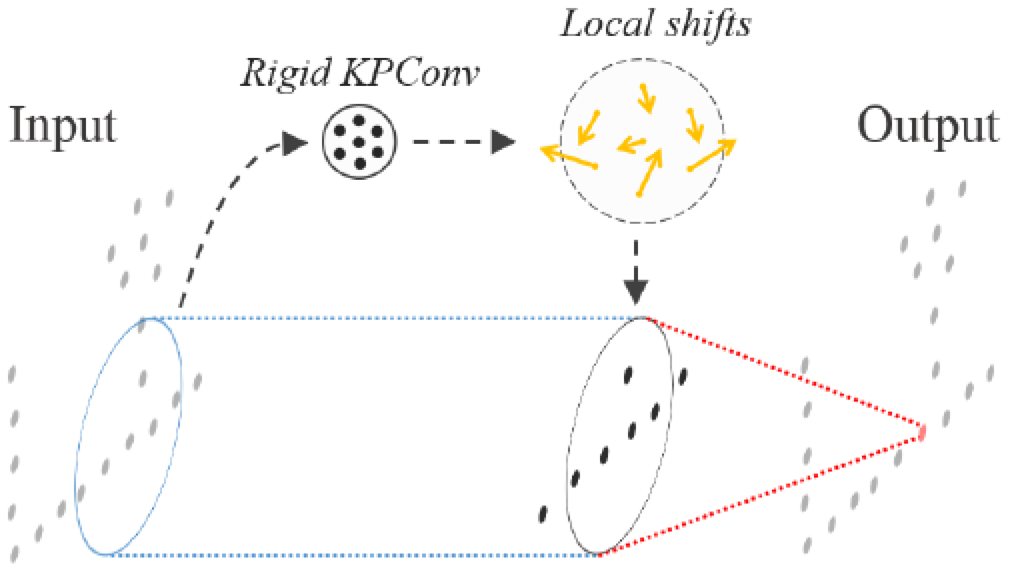
$$F_{c_{out}} = \sum_{c_{in}} \sum_{x_i \in R_k} \sum_{k} W_k^{c_{in}} h(p_k, x_i) F_{x_i}^{c_{in}}$$

$$h\left(y_i, \widetilde{x}_k\right) = \max\left(0, 1 - \frac{\|y_i - \widetilde{x}_k\|}{\sigma}\right)$$

Kernel Position

- Deformable Kernel

$$g_{deform}(y_i, \Delta(x)) = \sum_{k<K} h\left(y_i, \widetilde{x}_k + \Delta_k(x)\right) W_k$$



3D 点云处理的理想性质

1. computationally efficient ?
2. invariant to the order of points in the point cloud √
3. robust to different samplings and varying densities ×(scale)
4. translation invariant ×(rotate)
5. ...

Experiment

	ModelNet40	ShapeNetPart	
Methods	OA	mcIoU	mIoU
SPLATNet [34]	-	83.7	85.4
SGPN [42]	-	82.8	85.8
3DmFV-Net [9]	91.6	81.0	84.3
SynSpecCNN [48]	-	82.0	84.7
RSNet [15]	-	81.4	84.9
SpecGCN [40]	91.5	-	85.4
PointNet++ [27]	90.7	81.9	85.1
SO-Net [19]	90.9	81.0	84.9
PCNN by Ext [2]	92.3	81.8	85.1
SpiderCNN [45]	90.5	82.4	85.3
MCConv [13]	90.9	-	85.9
FlexConv [10]	90.2	84.7	85.0
PointCNN [20]	92.2	84.6	86.1
DGCNN [43]	92.2	85.0	84.7
SubSparseCNN [9]	-	83.3	86.0
KPConv *rigid*	**92.9**	85.0	86.2
KPConv *deform*	92.7	**85.1**	**86.4**

Methods	Scannet	Sem3D	S3DIS	PL3D
Pointnet [26]	-	-	41.1	-
Pointnet++ [27]	33.9	-	-	-
SnapNet [4]	-	59.1	-	-
SPLATNet [34]	39.3	-	-	-
SegCloud [37]	-	61.3	48.9	-
RF_MSSF [38]	-	62.7	49.8	56.3
Eff3DConv [50]	-	-	51.8	-
TangentConv [36]	43.8	-	52.6	-
MSDVN [30]	-	65.3	54.7	66.9
RSNet [15]	-	-	56.5	-
FCPN [28]	44.7	-	-	-
PointCNN [20]	45.8	-	57.3	-
PCNN [2]	49.8	-	-	-
SPGraph [17]	-	73.2	58.0	-
ParamConv [41]	-	-	58.3	-
SubSparseCNN [9]	**72.5**	-	-	-
KPConv *rigid*	68.6	**74.6**	65.4	72.3
KPConv *deform*	68.4	73.1	**67.1**	**75.9**