Tsinghua University

i-VisionGroup

文献分享：
Flex-conv
KPconv

刘宗健
8.27

目录

□ 背景

□ 卷积方法

□ 网络层

□ 实验

Flex-Convolution Million-Scale Point-Cloud Learning Beyond Grid-Worlds

Kpconv: Flexible and deformable convolution for point clouds. ICCV 2019. Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Franc ¸ois Goulette, and Leonidas J Guibas.

背景

- 卷积网络从图像到点云的困境

  数据存储不规则（像素vs位置信息）

  无序性

  稀疏性

背景

□ 与网格的联系

空间中可以局域化定位

点 $P \in R^{N*3}$ 视为结构要素
特征 $F \in R^{N*D}$ 视为实际数据

最大分辨率的限制

背景

□ 点云的处理思路

1. 套用CNN

   多个视角二维图像；三维网格化

2. 直接点云输入

   对称函数处理（以pointnet为典型的MLP网络）；图卷积网络

3. 利用点云特性直接卷积操作

   PointwiseCNN，SpiderCNN，PCNN，Flex-conv

卷积方法1

$$(w \circledast f)[\ell] = \sum_{c \in C} \sum_{\tau \in \{-1,0,1\}^2} w_{c'}(c,\tau) f(c, \ell - \tau),$$

$$P = \left\{ (\ell^{(i)}, f^{(i)}) \in L \times F \mid i = 0, 1, \ldots, n-1 \right\}.$$

Nkजझष ठज़, lझ़, . . . , l′k−झ़ड

$$f'(c', \ell^{(i)}) = \sum_{c \in C} \sum_{\ell' \in \mathcal{N}_k(\ell^{(i)})} \tilde{w}(c, \ell^{(i)}, \ell') \cdot f(c, \ell').$$

$$\tilde{w}_{c'} : C \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, \quad (c, \ell, \ell') \mapsto \tilde{w}(c, \ell, \ell')$$

□ 标准数积

$$\tilde{w}(c,\ell,\ell' | \theta_c,\theta_{b_c}) = \langle \theta_c, \ell - \ell' \rangle + \theta_{b_c}\,.$$

□ 处处定义良好，连续可微

核卷积定义2

- □ xi 是点 P ∈ $R^{N\times3}$

  fi 是其特征 F ∈ $R^{N\times D}$

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x) f_i$$

$$\{\ x_i \in \mathcal{P} \mid \|x_i - x\| \leqslant r\ \}$$

Nx ष़ $y_i = x_i - x$

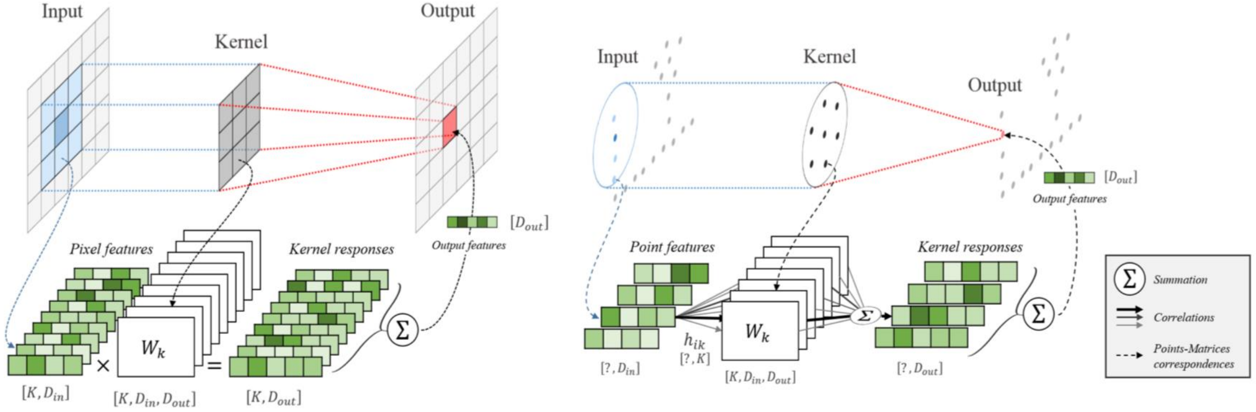
$$\mathcal{B}_r^3 = \{y \in \mathbb{R}^3 \mid \|y\| \leqslant r\}$$

- Kernel points(核点)

$$\{\widetilde{x}_k \mid k < K\} \subset \mathcal{B}_r^3 \qquad \mathcal{B}_r^3 = \{y \in \mathbb{R}^3 \mid \|y\| \leqslant r\}$$

$$\{W_k \mid k < K\} \subset \mathbb{R}^{D_{in} \times D_{out}}$$

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x) f_i \qquad g(y_i) = \sum_{k<K} h\left(y_i, \widetilde{x}_k\right) W_k$$



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

□ 核点的初始化产生

优化问题：k个点尽量远离（每个点分配排斥势）

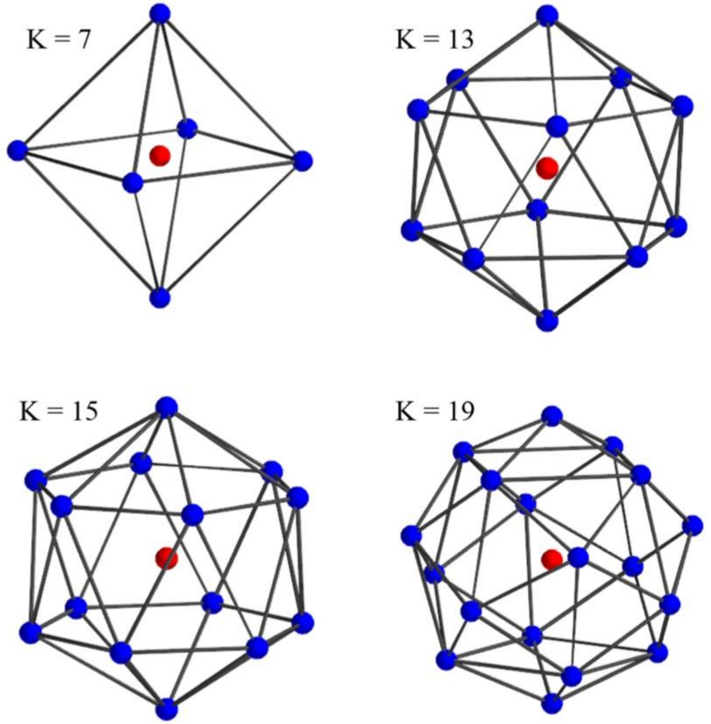
$$\forall x \in \mathbb{R}^3, \quad E_k^{rep}(x) = \frac{1}{\|x - \widetilde{x}_k\|}$$

避免无限发散（中心加引力势 ）

$$\forall x \in \mathbb{R}^3, \quad E^{att}(x) = \|x\|^2$$

能量最小化

$$E^{tot} = \sum_{k<K} \left(E^{att}(\widetilde{x}_k) + \sum_{l \neq k} E_k^{rep}(\widetilde{x}_l) \right)$$



□ 刚性or可变形核

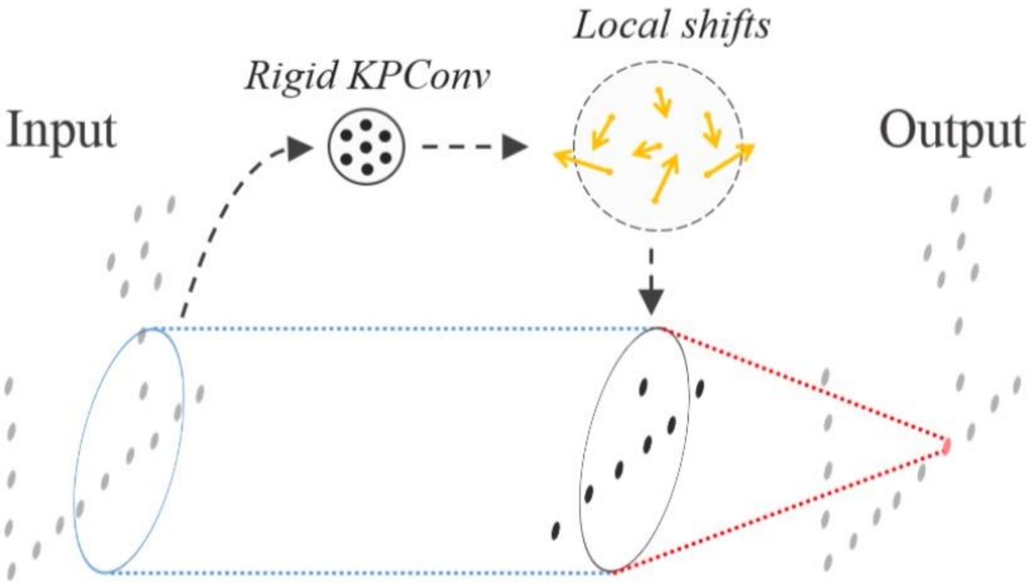
$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g_{deform}(x - x_i, \Delta(x)) f_i$$

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

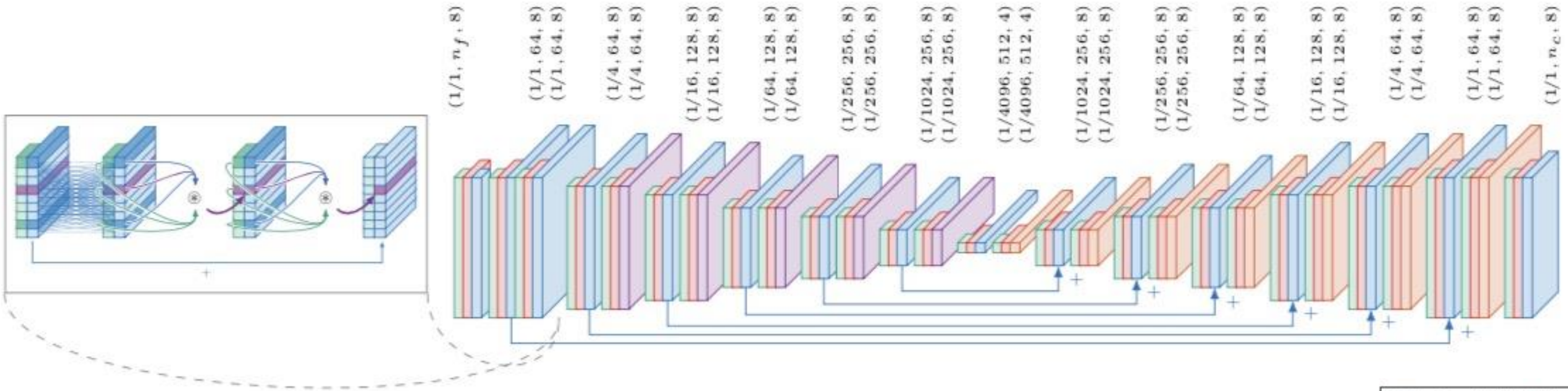
$$\mathcal{L}_{reg} = \sum_{x} \mathcal{L}_{fit}(x) + \mathcal{L}_{rep}(x)$$

$$\mathcal{L}_{fit}(x) = \sum_{k<K} \min_{y_i} \left(\frac{\|y_i - (\widetilde{x}_k + \Delta_k(x))\|}{\sigma} \right)^2$$

$$\mathcal{L}_{rep}(x) = \sum_{k<K} \sum_{l \neq k} h\left(\widetilde{x}_k + \Delta_k(x), \widetilde{x}_l + \Delta_l(x)\right)^2$$



网络结构1



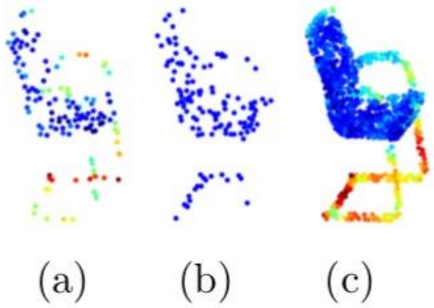
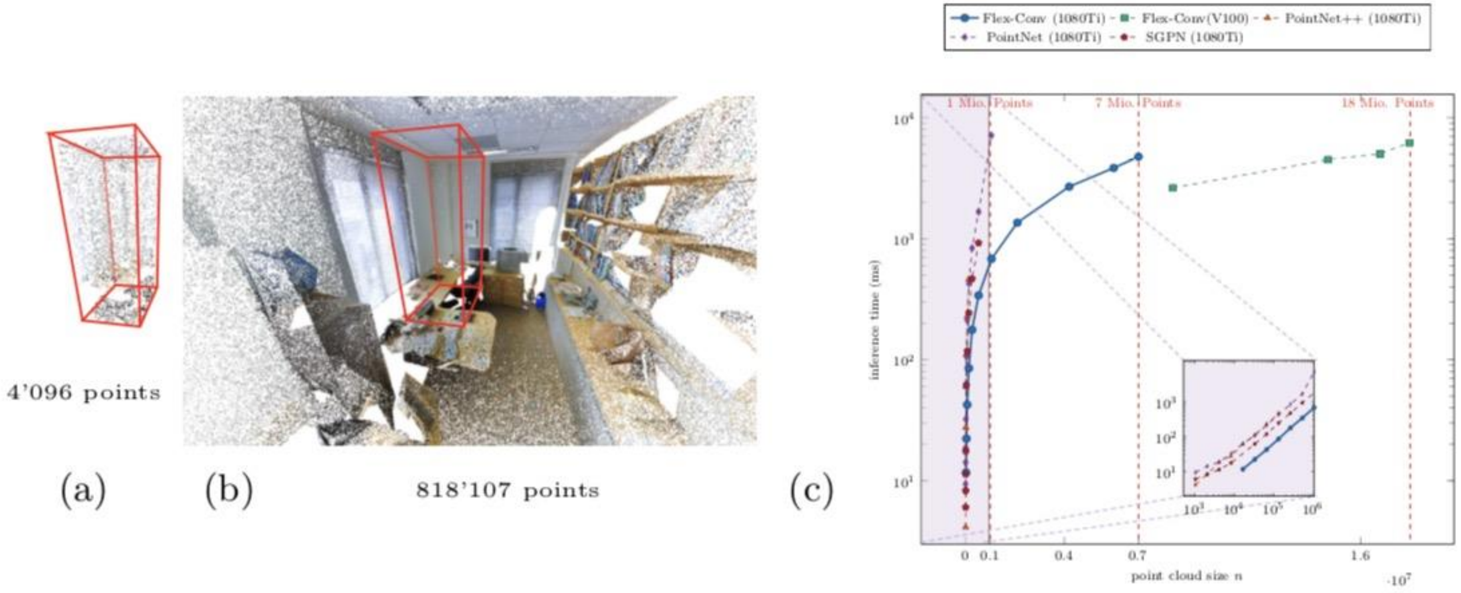


Fig. 3: IDISS (a) against random sub-sampling (b) for an object (c) with color-coded density.



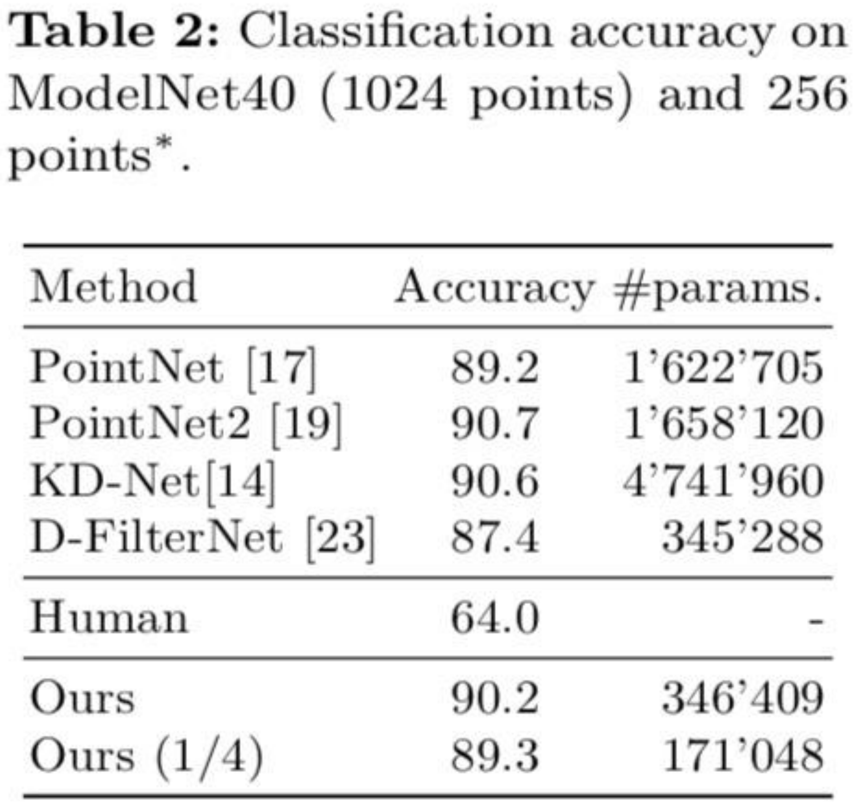
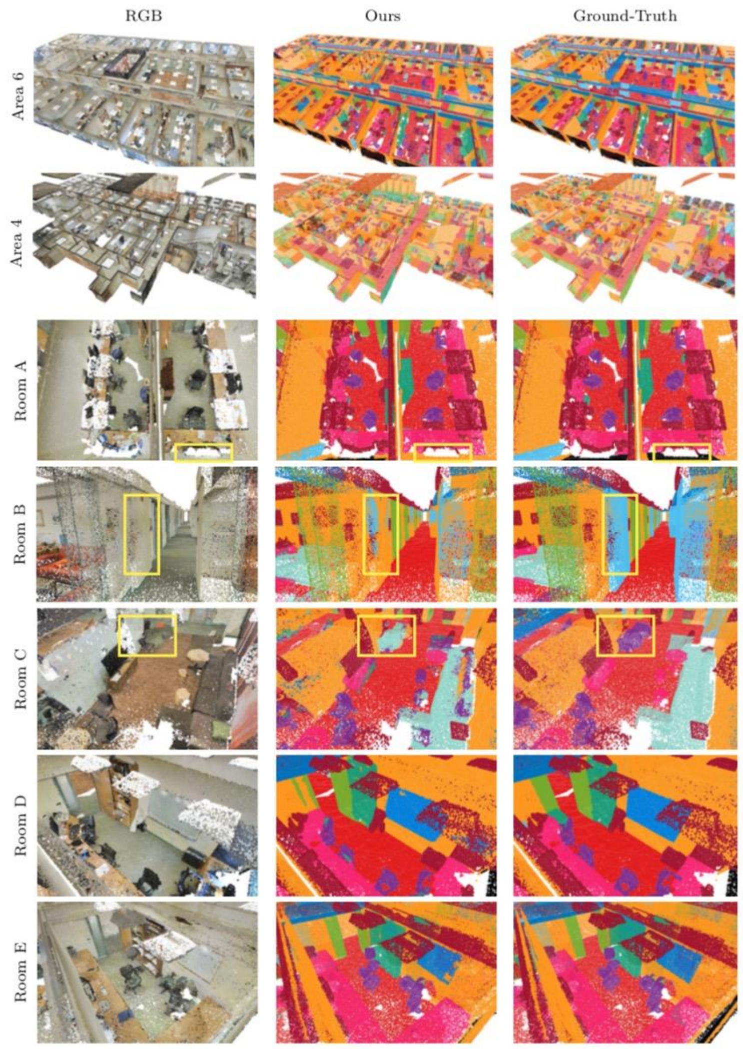


Table 2: Classification accuracy on ModelNet40 (1024 points) and 256 points*.

Method	Accuracy	#params.
PointNet [17]	89.2	1’622’705
PointNet2 [19]	90.7	1’658’120
KD-Net[14]	90.6	4’741’960
D-FilterNet [23]	87.4	345’288
Human	64.0	-
Ours	90.2	346’409
Ours (1/4)	89.3	171’048



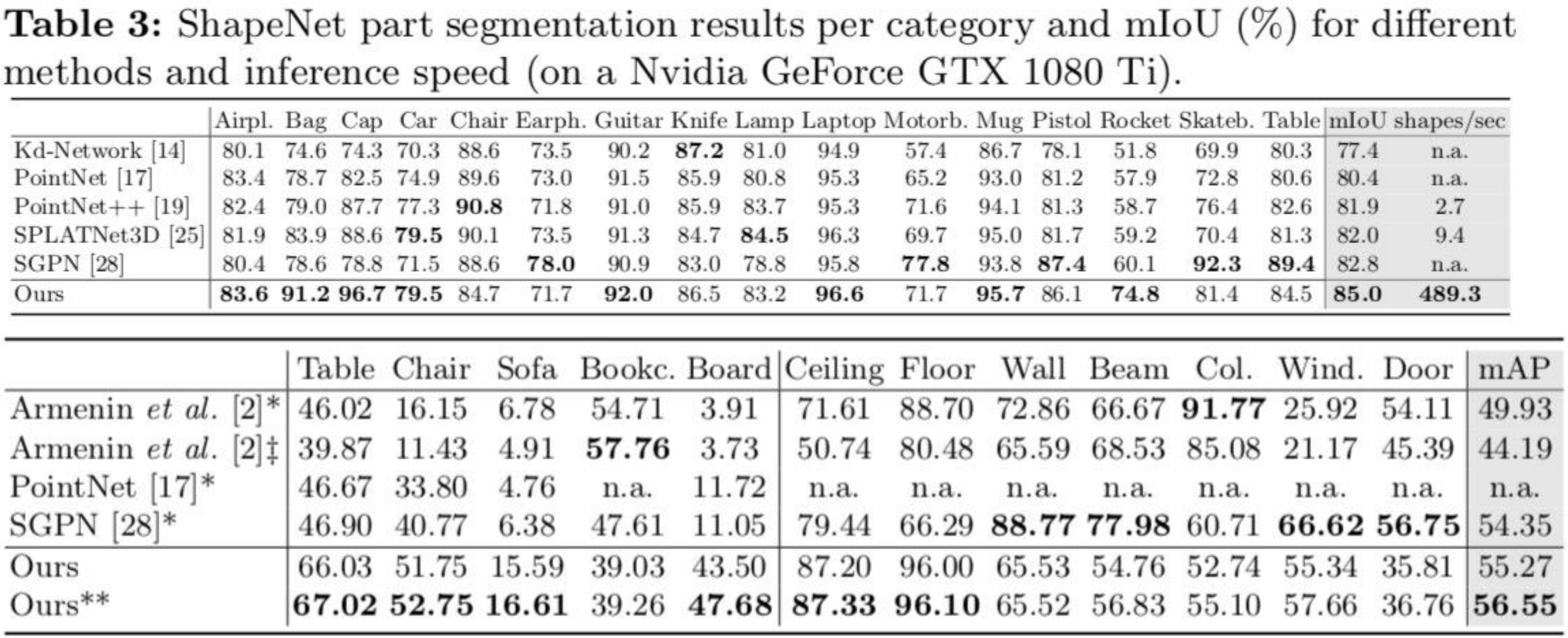


Table 3: ShapeNet part segmentation results per category and mIoU (%) for different methods and inference speed (on a Nvidia GeForce GTX 1080 Ti).

	Airpl.	Bag	Cap	Car	Chair	Earph.	Guitar	Knife	Lamp	Laptop	Motorb.	Mug	Pistol	Rocket	Skateb.	Table	mIoU	shapes/sec
Kd-Network [14]	80.1	74.6	74.3	70.3	88.6	73.5	90.2	**87.2**	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3	77.4	n.a.
PointNet [17]	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6	80.4	n.a.
PointNet++ [19]	82.4	79.0	87.7	77.3	**90.8**	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6	81.9	2.7
SPLATNet3D [25]	81.9	83.9	88.6	**79.5**	90.1	73.5	91.3	84.7	**84.5**	96.3	69.7	95.0	81.7	59.2	70.4	81.3	82.0	9.4
SGPN [28]	80.4	78.6	78.8	71.5	88.6	**78.0**	90.9	83.0	78.8	95.8	**77.8**	93.8	**87.4**	60.1	**92.3**	**89.4**	82.8	n.a.
Ours	**83.6**	**91.2**	**96.7**	**79.5**	84.7	71.7	**92.0**	86.5	83.2	**96.6**	71.7	**95.7**	86.1	**74.8**	81.4	84.5	**85.0**	**489.3**

	Table	Chair	Sofa	Bookc.	Board	Ceiling	Floor	Wall	Beam	Col.	Wind.	Door	mAP
Armenin *et al.* [2]*	46.02	16.15	6.78	54.71	3.91	71.61	88.70	72.86	66.67	**91.77**	25.92	54.11	49.93
Armenin *et al.* [2]‡	39.87	11.43	4.91	**57.76**	3.73	50.74	80.48	65.59	68.53	85.08	21.17	45.39	44.19
PointNet [17]*	46.67	33.80	4.76	n.a.	11.72	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
SGPN [28]*	46.90	40.77	6.38	47.61	11.05	79.44	66.29	**88.77**	**77.98**	60.71	**66.62**	**56.75**	54.35
Ours	66.03	51.75	15.59	39.03	43.50	87.20	96.00	65.53	54.76	52.74	55.34	35.81	55.27
Ours**	**67.02**	**52.75**	**16.61**	39.26	**47.68**	**87.33**	**96.10**	65.52	56.83	55.10	57.66	36.76	**56.55**

网络结构2

- □ 采样层
- □ 池化层
- □ KPConv层
- □ 网络参数

每一层有单元格大小dl_j

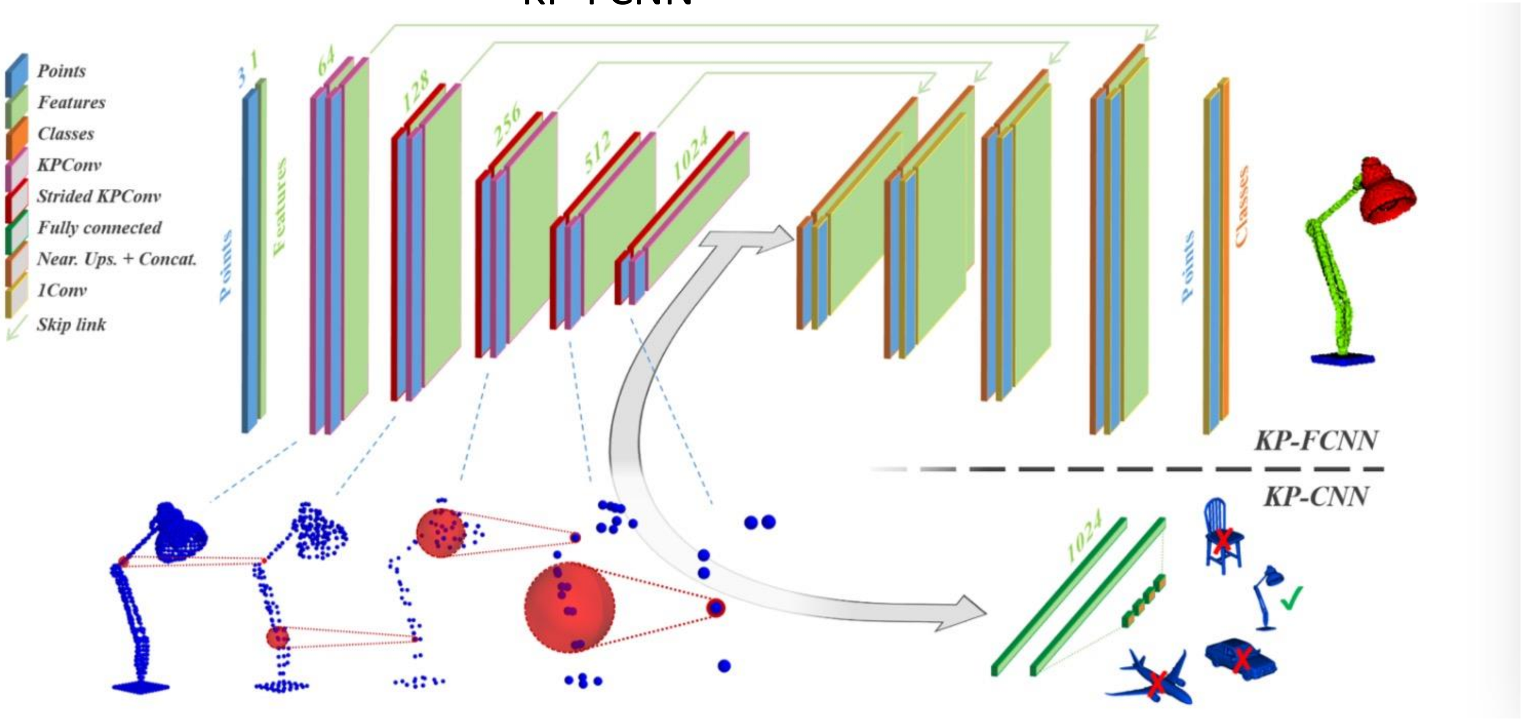
$$\sigma_j = \Sigma \times dl_j$$

$$r_j = \rho \times dl_j$$

$$dl_{j+1} = 2 * dl_j.$$

$$K = 15, \Sigma = 1.0 \text{ and } \rho = 5.0.$$

KP-FCNN



KP-CNN

实验

□ 场景分割

Scannet(室内场景，20类，1513个训练场景)

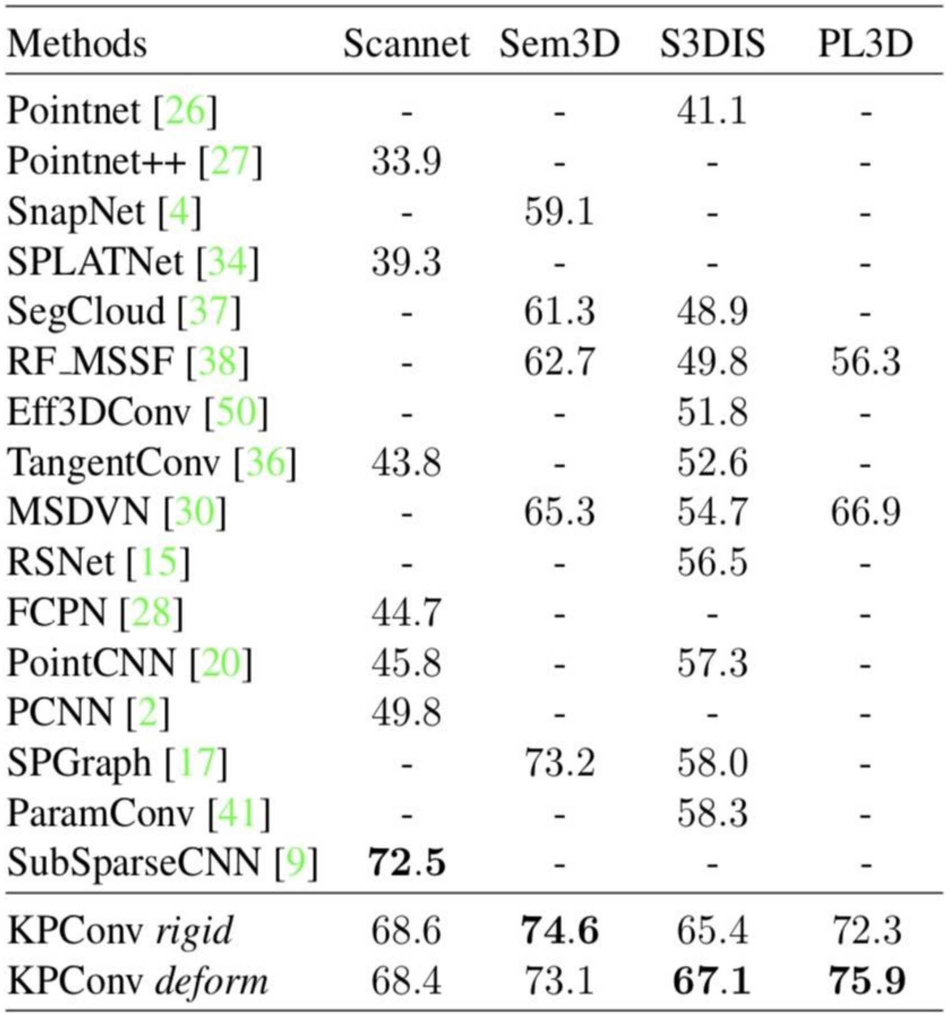
S3DIS（6个大型室内，13类，2.73亿点）

Semantic3D（在线基准，户外场景，40亿点8类）

Paris-Lille-3D(4个城市2km街道，1.6亿10类)

KP-FCNN 随机选择球训练和测试

输入球半径选择为50×dl_0



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**

□ 分割结果

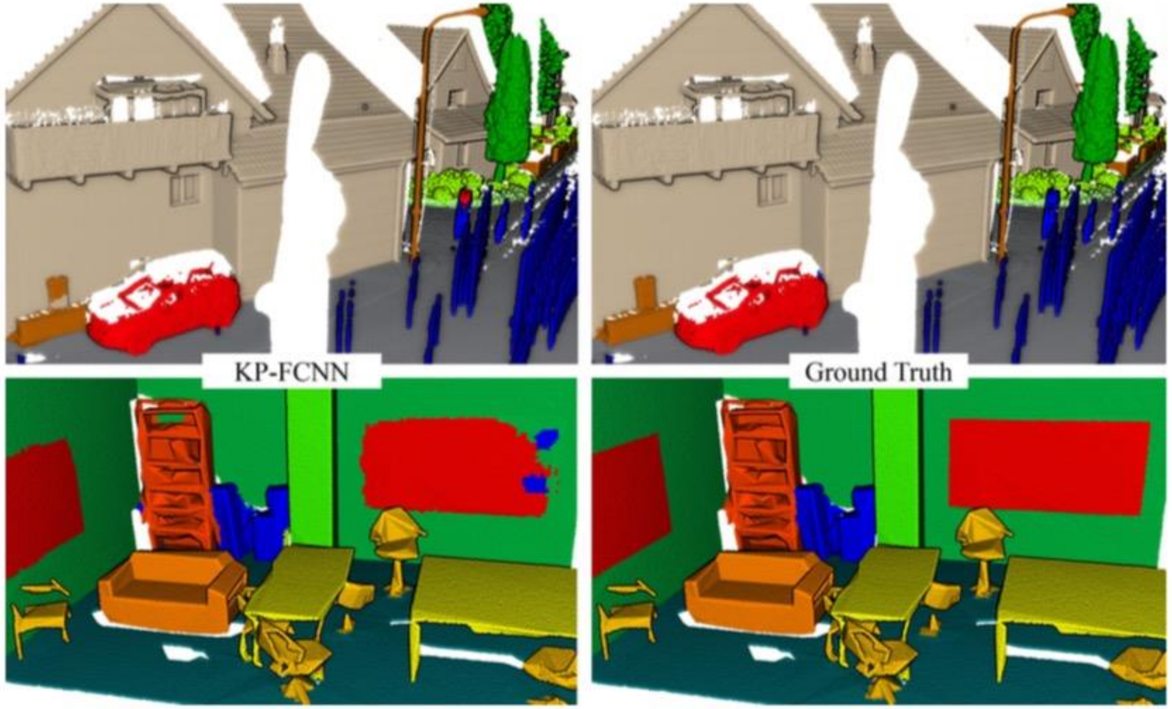
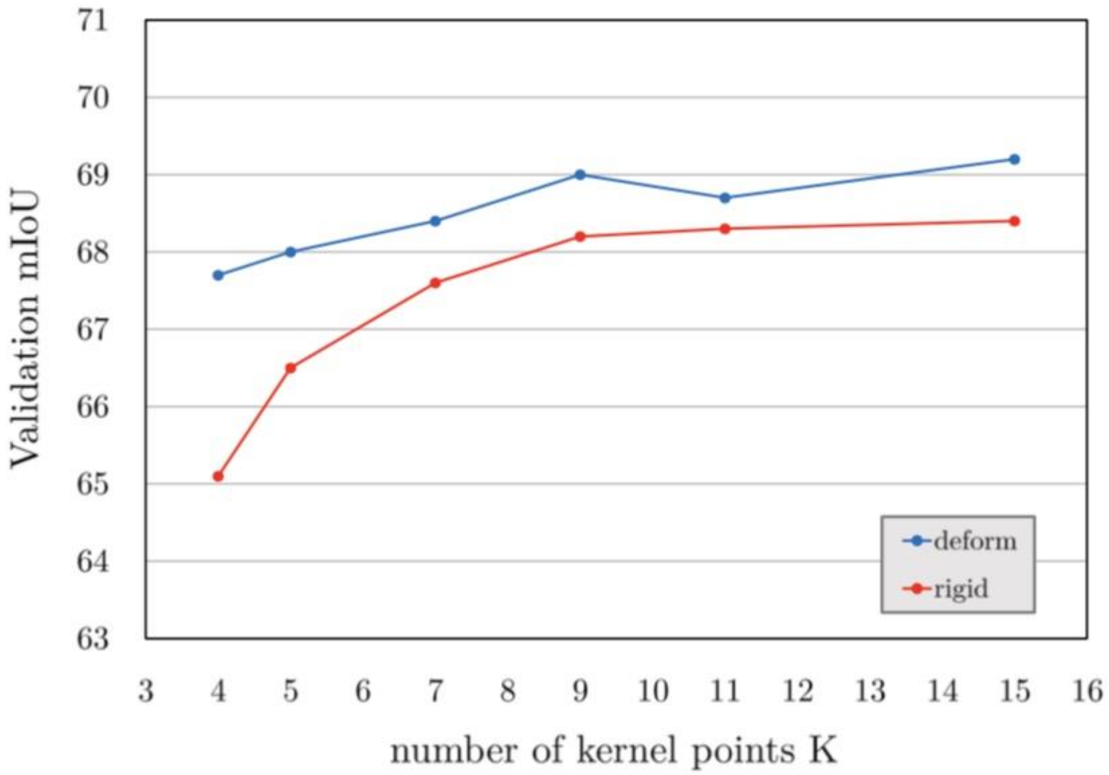


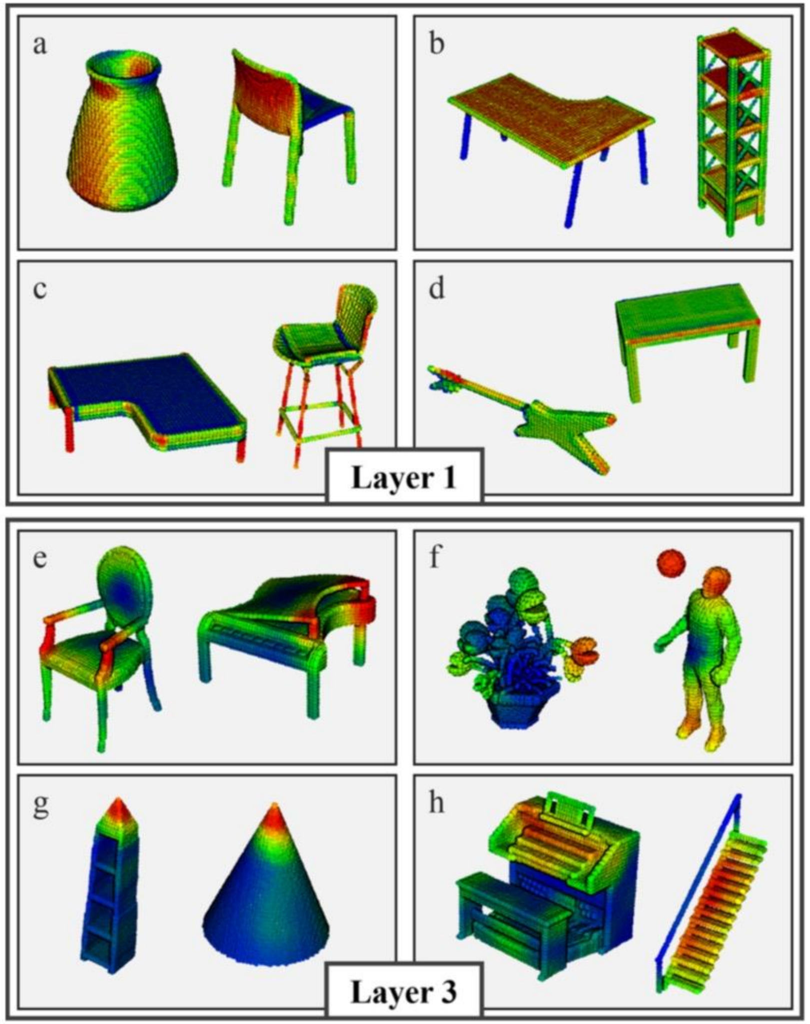
Figure 4. Outdoor and Indoor scenes, respectively from Semantic3D and S3DIS, classified by KP-FCNN with deformable kernels.

□ 消融实验

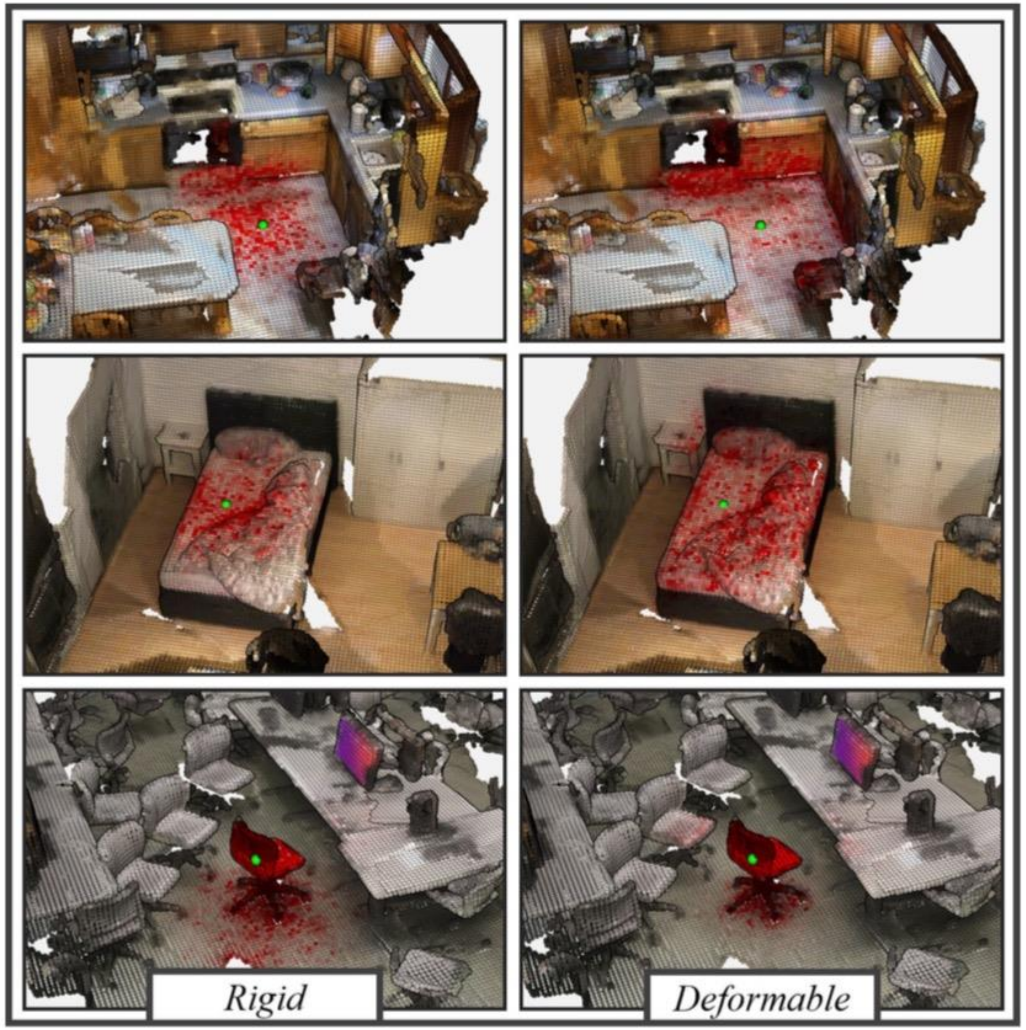


71
70
69
68
67
66
65
64
63
Validation mIoU
3
4
5
6
7
8
9
10
11
12
13
14
15
16
number of kernel points K
deform
rigid

□ 特征可视化



□ 有效感受野



谢谢聆听！