

PointNet 文献阅读及拓展阅读

原创 万俟淋曦 2020-03-27 12:09:28 462 收藏 4 原力计划 版权

分类专栏: 深度学习 文献阅读 文章标签: 计算机视觉 神经网络 点云 PointNet 深度学习

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

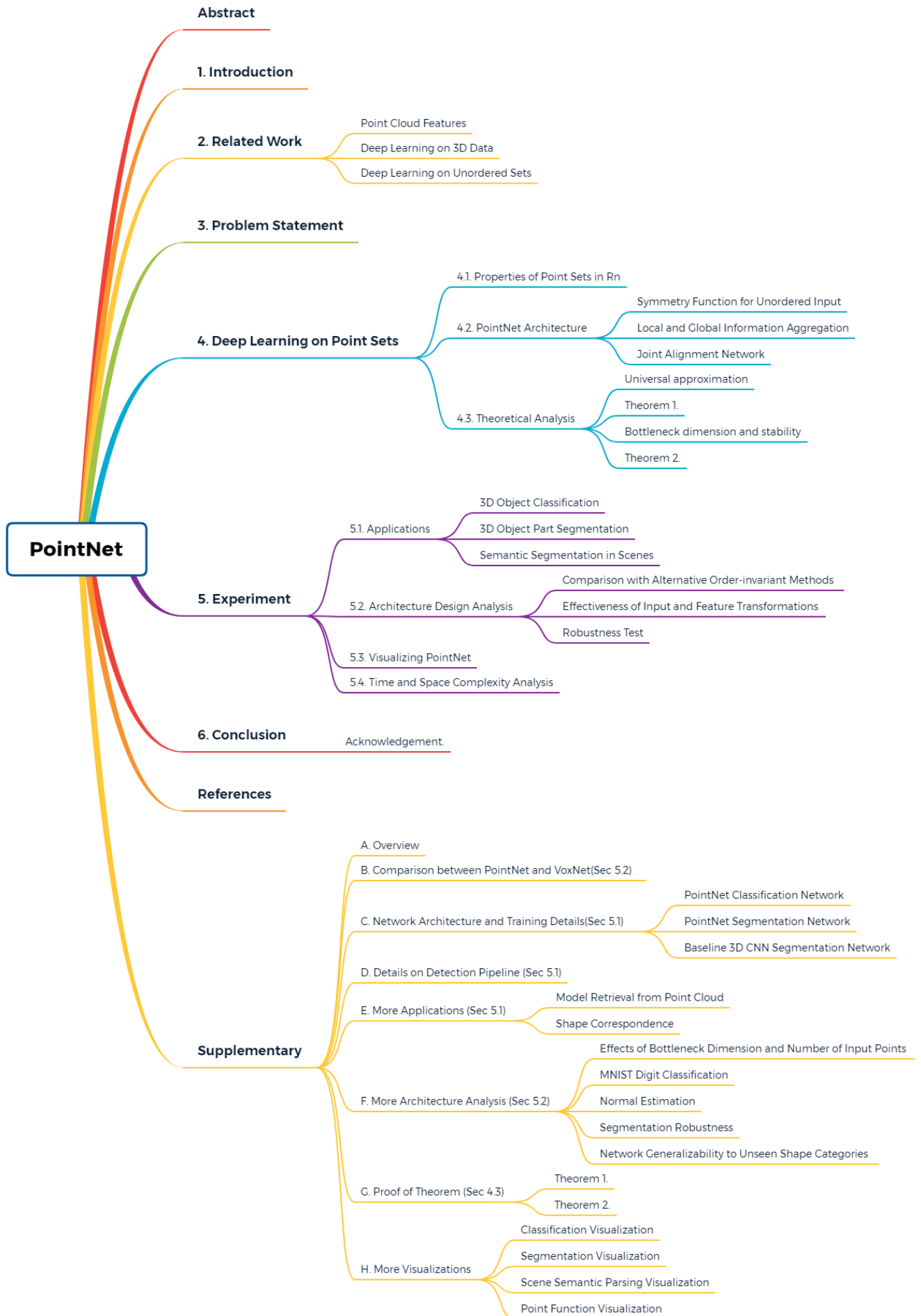
Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas
Stanford University

一、概览

论文地址: <https://arxiv.org/pdf/1612.00593.pdf>

代码下载: <https://github.com/charlesq34/pointnet>

论文框架:



研究目的： 利用深度学习网络处理三维点云数据。

研究思路： 首先通过对齐网络保证点云对特定空间转换的不变性，然后将全局特征和局部特征进行串联，综合利用特征。

解决方法： 提出PointNet —— 一种直接对三维几何数据(如点云或网格)进行处理的深度学习模型。

研究贡献：

- 1. 设计了一种直接输入三维无序点集的新型深度网络体系结构。
- 2. 解决点云对特定空间转换的不变性。
- 3. 展示了如何训练这样的网络来执行3D形状分类、形状部分分割和场景语义分析任务。

实验数据集： ModelNet40、ShapeNet、Stanford 3D semantic parsing data set

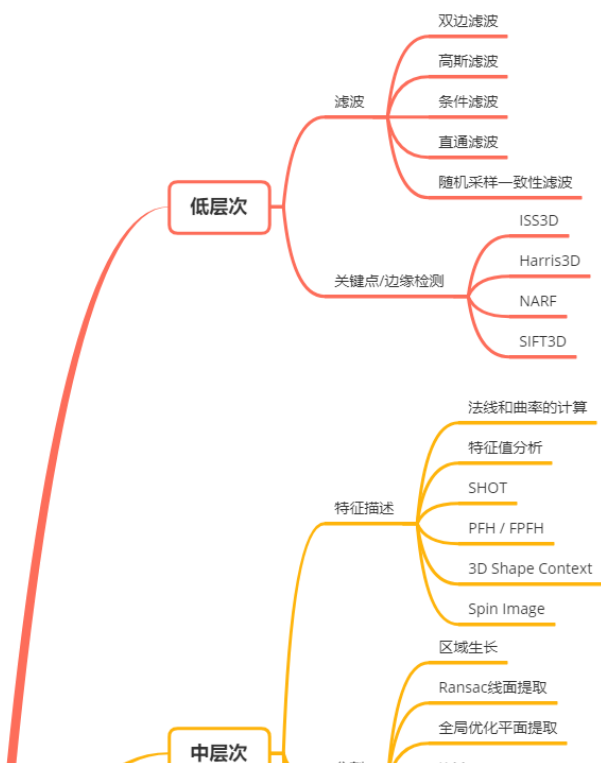
二、关于点云

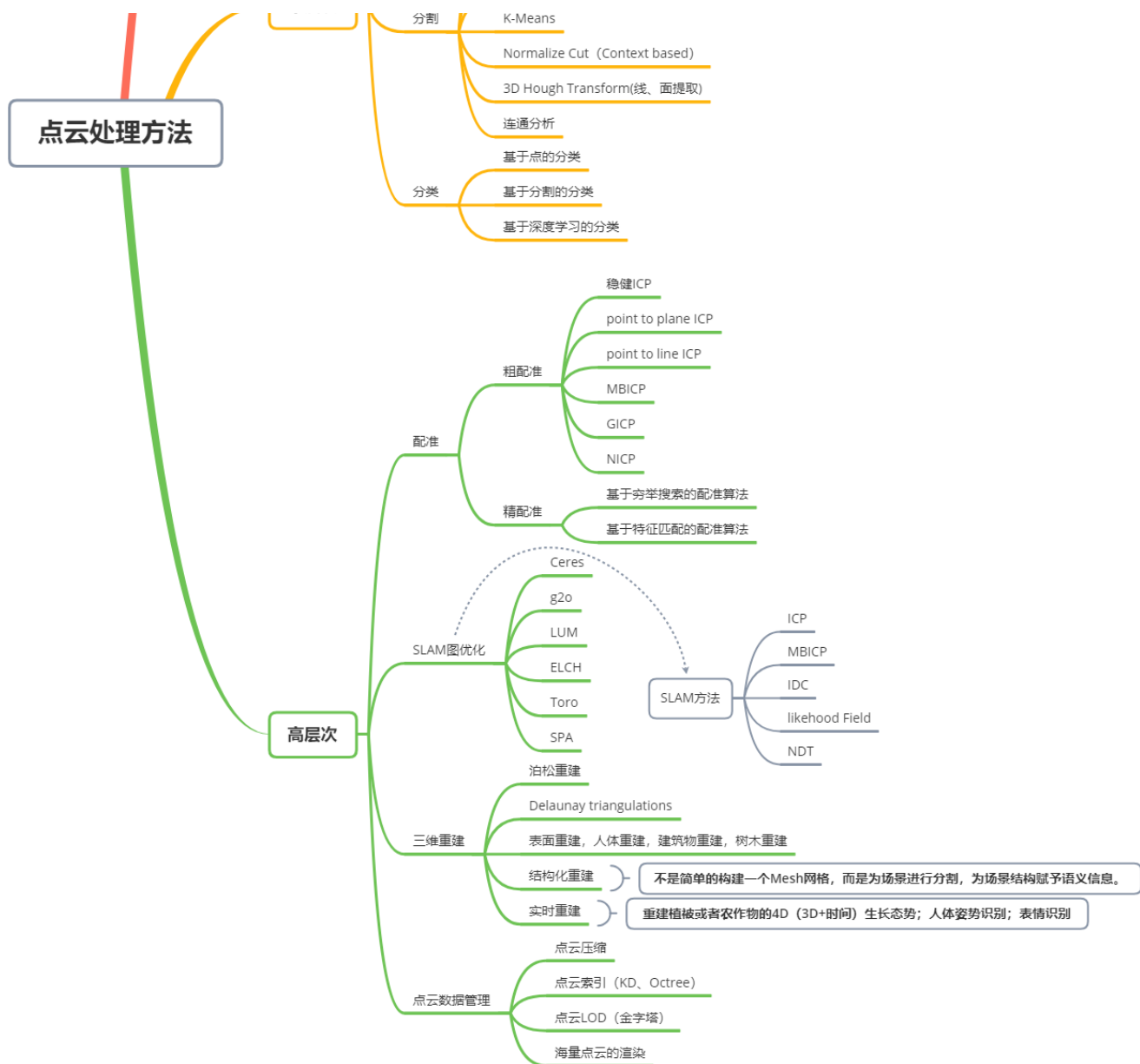
点云的概念： 点云是在同一空间参考系下表达目标空间分布和目标表面特性的海量点集合，包含了丰富的信息，可以是三维坐标X, Y, Z、颜色、强度值、时间等等。点云本质上是一长串点（Nx3矩阵，其中n是点数）。

点云包含的信息：

- 1. 根据激光测量原理得到的点云，包括三维坐标（XYZ）和激光反射强度（Intensity），强度信息与目标的表面材质、粗糙度、入射角方向，以及仪器的发射能量，激光波长有关。
- 2. 根据摄影测量原理得到的点云，包括三维坐标（XYZ）和颜色信息（RGB）。
- 3. 结合激光测量和摄影测量原理得到点云，包括三维坐标（XYZ）、激光反射强度（Intensity）和颜色信息（RGB）。

点云处理方法：





<https://blog.csdn.net/maizousidemao>

点云数据是在欧式空间下的点的一个子集，它具有以下三个特征：

2.1. 无序性

与图像中的像素阵列或体积网格中的体素阵列不同，点云是一组没有特定顺序的点。

点云数据是一个集合，对数据的顺序是不敏感的。这就意味着处理点云数据的模型需要对数据的不同排列保持不变性。

目前文献中使用的方法包括：

1. 将无序的数据重排序。
2. 用数据的所有排列进行数据增强然后使用RNN模型。
3. 用对称函数来保证排列不变性。

由于第三种方式的简洁性且容易在模型中实现，论文作者选择使用第三种方式，既使用maxpooling 这个对称函数来提取点云数据的特征。

2.2. 点与点之间的空间关系

一个物体通常由特定空间内一定数量的点云构成，也就是说这些点云之间存在着空间关系。因此，模型需要能够捕捉附近点的局部结构，以及局部结构之间的组合相互作用。为了能有效利用这种空间关系，论文作者提出了将局部特征和全局特征进行串联的方式来聚合信息。

2.3. 刚性变换不变性

点云数据所代表的目标对某些空间变换应该具有不变性，如旋转和平移等刚性变换。论文作者提出了在进行特征提取之前，先对点云数据进行对齐的方式来保证不变性。对齐操作是通过训练一个小型的网络来得到变换矩阵，并将之和输入点云数据相乘来实现。

三、现阶段研究

3.1. Point Cloud Features

大多数现有的点云功能都是针对特定任务手工创建的。点特征通常编码某些点的统计属性，并且被设计成对某些变换是不变的，这些变换通常被分类为内部的或外部的。它们还可以分为局部特征和全局特征。对于特定的任务，要找到最优的特征组合并不是一件容易的事。

3.2. Deep Learning on 3D Data

根据3D数据的表示不同有多种深度学习的方法

Volumetric CNNs: 通过将物体表现为空间中的体素进行类似于二维的三维卷积。然而，由于三维卷积的数据稀疏性和计算开销，体素表示受到其时间和空间复杂度的限制，对于处理非常大的点云具有挑战性，目前已经不是主流的方法了。

Multiview CNNs: 通过多视角二维图片组合为三维物体，此方法将传统CNN应用于多张二维视角的图片，特征被view pooling procedure聚合起来形成三维物体。

Spectral CNNs: 在网格上使用谱CNNs。然而，这些方法目前仅限于有机对象等多种网格上，如何将其扩展到非等轴测形状(如家具)上并不明显。

Feature-based DNNs: 首先通过提取传统的形状特征将三维数据转化为向量，然后使用全连通网络对形状进行分类。我们认为它们受到所提取特征的表征能力的限制。

3.3. Deep Learning on Unordered Sets

从数据结构的角度来看，点云是一组无序的矢量。虽然大多数深度学习的研究都集中在规则的输入上，如序列(语音和语言处理)、图像和体积(视频或3D数据)，但在点集的深度学习方面却没有做太多的工作。

Oriol Vinyals等人最近的一项研究探讨了这个问题。他们使用一个带有注意力机制的 read-process-write 网络来消费无序的输入集，并表明他们的网络具有对数字进行排序的能力。然而，由于他们的工作集中在一般集合和NLP应用上，在集合中缺少几何学的作用。

四、本文方法

点云表示为一组三维点 $\{P_i \mid i = 1, \dots, n\}$ ，其中每个点 P_i 都是 (x, y, z) 坐标的向量加上额外的功能通道(如颜色、法线等)。为简单和清晰起见，除非另有说明，否则我们仅使用 (x, y, z) 坐标作为点的表示。

我们网络架构的灵感来自于 R^n 中的点集属性。

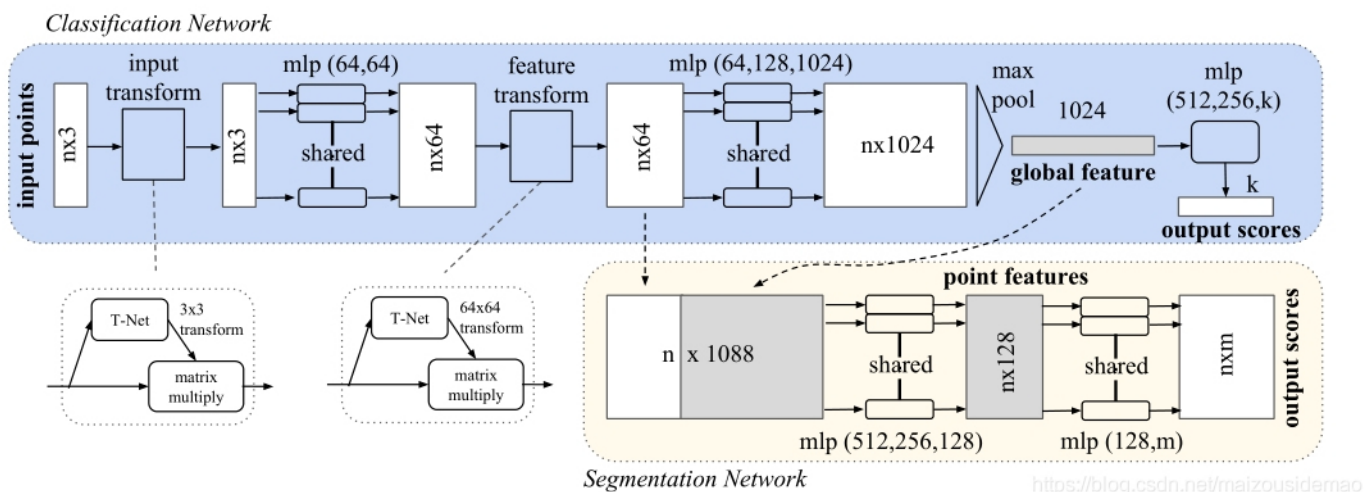
4.1. R^n 中点集的属性 (Properties of Point Sets in R^n)

我们的输入是来自欧几里得空间的点的子集。它有三个主要属性：

1. 无序性

2. 点之间的空间关系
3. 刚性变换不变性
(此部分已在“二、关于点云”中详细说明)

4.2. PointNet 架构 (PointNet Architecture)



原文说明:

该分类网络 (Classification Network, 图中蓝色部分) 以 n 个点为输入, 应用输入和特征变换, 然后通过最大池化 (max pool) 的方法聚合点特征。输出是 k 个类的分类分数。分段网络 (Segmentation Network, 图中浅黄色部分) 是分类网络的扩展。它将全局和局部特征以及每类分数的输出连接在一起。

“MLP” 代表多层感知机, 括号中的数字为层大小。Batchnorm用于具有RELU的所有层。丢弃层用于分类网络中的最后一个MLP。

网络主要流程为:

1. 输入为一帧的全部点云数据的集合, 表示为一个 $n \times 3$ 的2d tensor, 其中 n 代表点云数量, 3对应xyz坐标。
2. 输入数据先通过和一个T-Net学习到的转换矩阵相乘来对齐 (input transform), 保证了模型对特定空间转换的不变性。
3. 通过多次mlp对各点云数据进行特征提取后, 再用一个T-Net对特征进行对齐 (feature transform)。
在特征的各个维度上执行最大池化 (max pool) 操作来得到最终的全局特征。
4. 对分类任务, 将全局特征通过mlp来预测最后的分类分数;
5. 对分割任务, 将全局特征和之前学习到的各点云的局部特征进行串联, 再通过mlp得到每个数据点的分类结果。

我们的网络有三个关键模块:

- 作为从所有点聚合信息的对称函数的最大池化层
- 局部和全局信息组合结构
- 两个同时对齐输入点和点要素的联合对齐网络

我们将在下面的单独段落中讨论这些设计选择背后的原因:

4.2.1 无序输入的对称函数 (Symmetry Function for Unordered Input)

该部分位于分类网络中（图中蓝色部分），构建分类网络主体。（个人理解）前面已经说过，对于无序输入，目前文献中使用的方法包括：

1. 将无序的数据重排序。
2. 用数据的所有排列进行数据增强然后使用RNN模型。
3. 用对称函数来保证排列不变性。

文章指出，重新排序的方法，在高维空间中对于点扰动并不稳定。而使用RNN的想法将点集视为顺序信号，并希望通过用随机排列的序列训练RNN，数据的顺序将按照输入顺序固定不变，OrderMatters 已经证明了顺序确实很重要，不能完全省略。虽然RNN对于长度较小(几十个)的序列的输入排序具有相对较好的鲁棒性，但很难扩展到数千个输入元素，而这是点集的常见大小。

为了使结果不受输入排列顺序的影响，本文作者提出通过对集合中的元素应用对称函数来近似定义在点集上的一般函数：

$$f(\{x_1, \dots, x_n\}) = g(h(x_1), \dots, h(x_n))$$

其中

$$f: 2^{\mathbb{R}^N} \rightarrow \mathbb{R}, h: \mathbb{R}^N \rightarrow \mathbb{R}^K, g: \underbrace{\mathbb{R}^K \times \dots \times \mathbb{R}^K}_n \rightarrow \mathbb{R}$$

g 是一个对称函数，用单变量函数和最大池化函数的组合来拟合。 h 用多层感知机网络（mlp）拟合，用于将点云数据映射到高维。

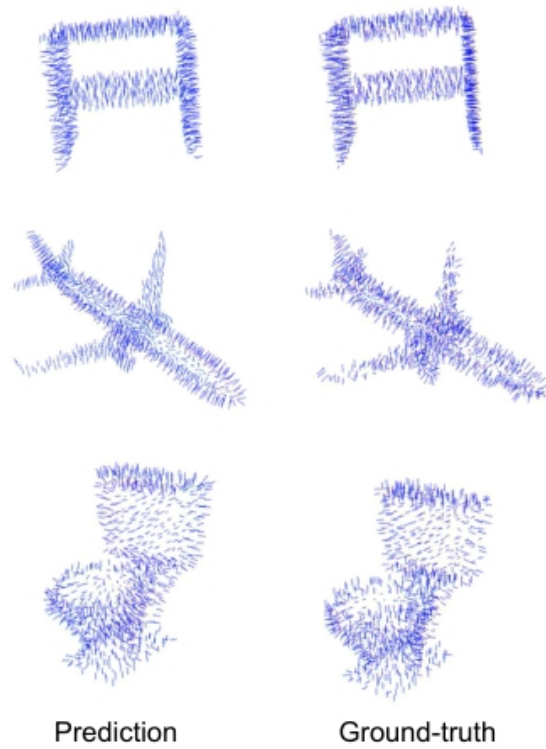
4.2.2 整合局部信息和全局信息 (Local and Global Information Aggregation)

该部分是分段网络（图中浅红色部分）。

上一步（分类网络）输出一个向量 $[f_1, \dots, f_k]$ ，是输入的点云数据的全局特征信息，在计算出全局点云特征向量后，通过将全局特征与每个点特征连接起来，再将其反馈给每个点特征。然后，基于组合点提取新的逐点特征，这样新的逐点特征同时知道局部和全局信息。

我们的网络能够预测依赖于局部几何和全局语义的每个点的数量。例如，我们可以准确地预测每个点的法线，验证网络能够汇总来自点的局部邻域信息。实验表明，该模型在形状分割和场景分割方面都达到了最好的效果。

PointNet 法线重建结果：



4.2.3 联合定位网络 (Joint Alignment Network)

如果点云经过某些几何变换(如刚性变换), 则点云的语义标签必须是不变的。因此, 我们期望通过我们的点集学习的表示对于这些变换是不变的。

自然的解决方案是在特征提取之前将所有输入集合与规范空间对齐。如 Max Jaderberg 等人的 Spatial Transformer Networks, 通过采样和插值来对齐2D图像, 并在GPU上实现了一个专门定制的层。我们通过一个微型网络 (T-net) 预测一个仿射变换矩阵, 并将这个变换直接应用到输入点的坐标上。T-net 本身类似于大网络, 由点无关特征提取、最大集合和全连通三个基本模块组成。第一个 T-net 以原始点云为输入, 回归为 3×3 矩阵的迷你点网(Mini-PointNet)。由每个点上的共享MLP(64, 128, 1024)网络(层输出大小为64, 128, 1024)、跨点的最大池和两个输出大小为512, 256的完全连接层组成。输出矩阵被初始化为单位矩阵。除最后一层外, 所有层都包含重新生成和批处理规格化。第二个 T-net 除了输出为 64×64 矩阵外, 网络结构与第一个相同。

这一思想也可以进一步推广到特征空间的对齐问题。我们可以在点要素上插入另一个对齐网络, 并预测一个特征转换矩阵来对齐来自不同输入点云的特征。然而, 特征空间中的变换矩阵比空间变换矩阵的维数要高得多, 极大地增加了优化的难度。因此, 我们将正则化项添加到 Softmax 训练损失中。我们将特征变换矩阵约束为接近正交矩阵:

$$L_{reg} = ||I - AA^T||_F^2$$

其中A是由 T-net 预测的特征对齐矩阵。我们发现, 通过添加正则项, 优化变得更加稳定, 我们的模型取得了更好的性能。

4.3. 理论分析 (Theoretical Analysis)

4.3.1 网络对函数的拟合能力

对于一个处理点云的网络, 通过集合函数的连续性, 对输入点集合的微小扰动应该不会对函数值有很大的改变, 例如分类或分割分数。

设 $\chi = \{S: S \subseteq [0, 1]^m \text{ and } |S| = n\}$, 而 $f: \chi \rightarrow \mathbb{R}$ 是 χ 上关于Hausdorff 距离 $d_H(\cdot, \cdot)$ 的连续集函数。即在m维欧式空间中, $\forall \epsilon > 0, \exists \delta > 0$, 对任意 $S, S' \in \chi$, 如果 $d_H(S, S') < \delta$, 则

$|f(S) - f(S')| < \epsilon$ 。如果在最大池化层有足够多的神经元，即 (2.1) 中公式的 K 足够大，PointNet 可以拟合任意的函数。

然后作者又给出两个定理：

Theorem 1.

设 $f: \mathcal{X} \rightarrow \mathbb{R}$ 是 \mathcal{X} 是关于 Hausdorff 距离 $d_H(\cdot, \cdot)$ 的连续集函数， $\forall \epsilon > 0$ ，存在连续函数 h 和对称函数 $g(x_1, \dots, x_n) = \gamma \circ \text{MAX}$ ，使得对任意 $S \in \mathcal{X}$ ，有：

(注： $f \circ g = f(g)$)

$$\left| f(S) - \gamma \left(\text{MAX}_{i=1, \dots, n} \{h(x_i)\} \right) \right|$$

$$= |f(S) - \gamma(\text{MAX}_{i=1, \dots, n} \{h(x_i)\})|$$

$< \epsilon$

$$\left| f(S) - \gamma \left(\text{MAX}_{i=1, \dots, n} \{h(x_i)\} \right) \right| < \epsilon$$

其中， x_1, \dots, x_n 是 S 中任意排序元素的完整列表， γ 是一个连续函数， MAX 是向量最大运算符，它接受 n 个向量作为输入，并返回一个包含各向量元素最大值的新向量。

这一定理主要是为了说明，PointNet 网络的表达能力受到最大池化层的维数的影响，即 (2.1) 中的 K ， K 越大网络表达能力越强。

4.3.2 网络的稳定性

下面的定理告诉我们，输入集中的小数据损坏或额外噪声点不太可能改变我们网络的输出：

Theorem 2.

假设 $u: \mathcal{X} \rightarrow \mathbb{R}^K$ 使得 $u = \text{MAX}_{i=1, \dots, n} \{h(x_i)\}$ ，并且 $f = \gamma \circ u$ ，那么，

- (a) 若 $C_S \subseteq T \subseteq N_S$ ，则 $\forall S, \exists C_S, N_S \subseteq \mathcal{X}, f(T) = f(S)$
- (b) $|C_S| \leq K$

(a) 这一定理表明，对于任何输入数据集 S ，都存在一个最小集 C_S 和一个最大集 N_S ，使得对 C_S 和 N_S 之间的任何集合 T ，其网络输出都和 S 一样。

(b) 说明了最小集 C_S 的数据多少由最大池化操作输出数据的维度 K 给出上界。换个角度来讲，PointNet 能够总结出表示某类物体形状的关键点，基于这些关键点 PointNet 能够判别物体的类别。这样的能力决定了 PointNet 对噪声和数据缺失的鲁棒性。

五、实验

实验分为四个部分。

1. 展示了 PointNets 可以应用于多个 3D 识别任务。
2. 提供了详细的实验来验证我们的网络设计。
3. 将网络学习的内容可视化。
4. 分析时间和空间复杂度。

5.1. 应用

5.1.1 三维对象分类

实验设计：

使用 ModelNet40 形状分类基准（数据集），包括来自 40 个人造物体类别的 12311 个 CAD 模型，分为 9843 个用于训练和 2468 个用于测试。

按照面面积对网格面上的1024个点进行均匀采样，并将其归一化为一个单位球面。在训练过程中，通过沿上轴（指向上方的坐标轴）随机旋转对象来动态增强点云，并用均值为0，标准差为0.02的高斯噪声抖动每个点的位置。

实验结果：

	input	#views	accuracy avg. class	accuracy overall
SPH [11]	mesh	-	68.2	-
3DShapeNets [28]	volume	1	77.3	84.7
VoxNet [17]	volume	12	83.0	85.9
Subvolume [18]	volume	20	86.0	89.2
LFD [28]	image	10	75.5	-
MVCNN [23]	image	80	90.1	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	89.2

虽然比大多数网络先进，但和基于多视图的方法(MVCNN)还有差距，作者认为这是由于丢失了渲染图像可以捕捉到的精细几何细节导致的。

5.1.2 三维物体零件分割

零件分割是一项具有挑战性的细粒度三维识别任务。给定3D扫描或网格模型，任务是为每个点或面分配零件类别标签(例如，椅腿、杯柄)。

实验设计：

我们在 ShapeNet 部件数据集上进行评估，该数据集包含16个类别的 16881 个形状，总共标注了 50 个部件。大多数对象类别都标有 2 到 5 个部分。

我们将零件分割表示为逐点分类问题。评价指标是基于点的 mIoU (mean Intersection over Union, 平均交并比)。比如，对于类别C的每个形状S，要计算形状的 mIoU 值：对于类别C中的每个部件，计算真实和预测之间的 IoU (Intersection over Union, 交并比)。如果真实和预测点的并集为空，则将部分 IoU 计数为 1。然后，我们对类别C中所有部分类型的 IoU 进行平均，以获得该形状的 mIoU。为了计算该类别的 mIoU，我们取该类别中所有形状的 mIoU 平均值。

我们将我们的分割版本 PointNet (架构图中 Segmentation Network 的修改版本)与两种传统方法 Interactive shape co-

segmentation via label propagation 和 A scalable active

framework for region annotation in 3d shape collections (这两种方法都利用了逐点几何特征和形状之间的对应关系)，以及我们自己的 3DCNN 进行了比较。

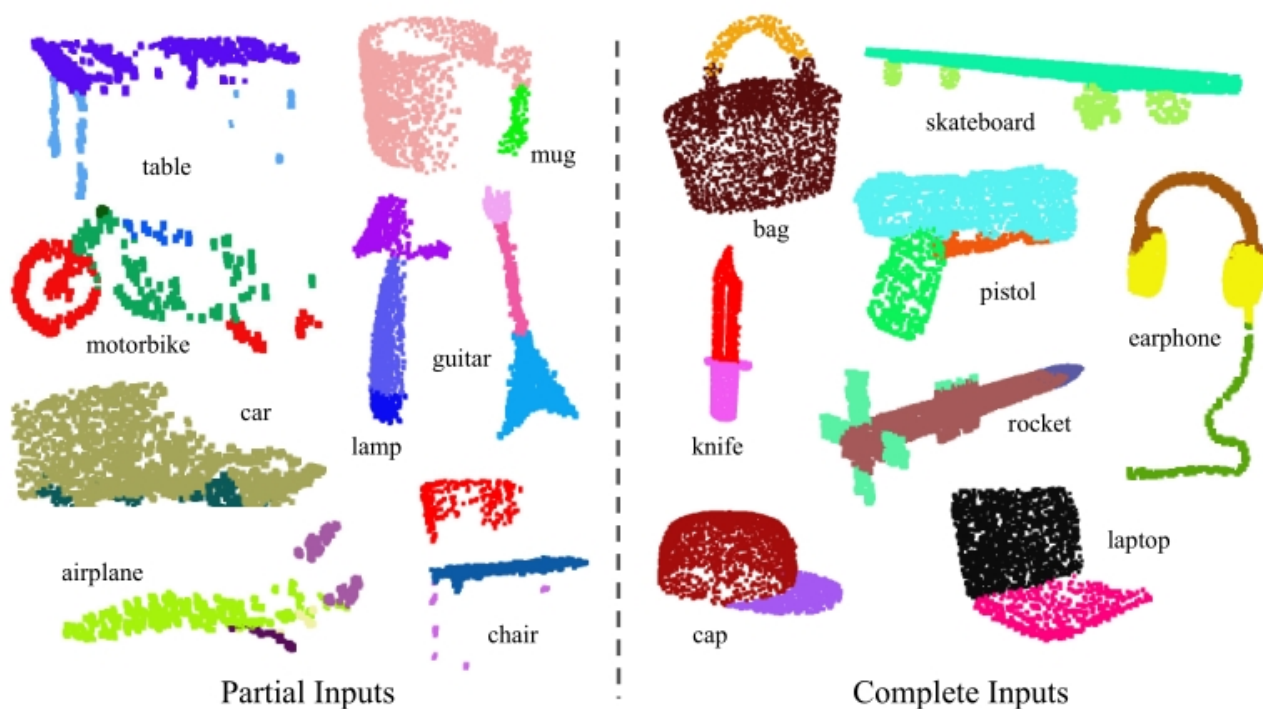
实验结果：

	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Wu [27]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [29]	81.4	81.0	78.4	77.7	75.7	87.6	61.9	92.0	85.4	82.5	95.7	70.6	91.9	85.9	53.1	69.8	75.3
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1
Ours	83.7	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

与 Yi 的方法相比，PointNet 的 mIoU 提高了 2.3%，并且在大多数类别中都优于 3DCNN。

补充实验：

还在模拟 Kinect 扫描上进行了实验，以测试这些方法的鲁棒性。对于 ShapeNet 零件数据集中的每一个 CAD 模型，我们使用 Blensor Kinect Simulator 从六个随机视角生成不完整的点云。并使用相同的网络架构和训练设置对 PointNet 进行完整形状和部分扫描的训练。结果如下：



mIoU 只下降了 5.3%。图中可以看出，虽然部分数据颇具挑战性，但我们的预测是合理的。

5.1.3 场景中的语义分割

我们的零件分割网络可以很容易地扩展到场景语义分割，其中点标签是语义对象类，而不是对象零件标签。

实验设计

我们在 Stanford 3D semantic parsing data set 进行了实验，该数据集包含来自 Matterport 扫描仪的 6 个区域(包括 271 个房间)的 3D 扫描。扫描中的每个点都用来自 13 个类别(椅子、桌子、地板、墙壁等加上杂物)中的一个语义标签进行注释。

为了准备训练数据，我们首先按房间划分点，然后将样本室分成面积为 $1\text{m} \times 1\text{m}$ 的块。我们训练我们的修改版本的 PointNet 来预测每个块中的每个点类。每个点由关于房间的 XYZ、RGB 和规格化位置(从 0 到 1)的 9 维向量表示。在训练时，我们在每个区块中动态随机抽样 4096 个点。在测试时，我们对所有的点进行测试。我们遵循与 3d semantic parsing of

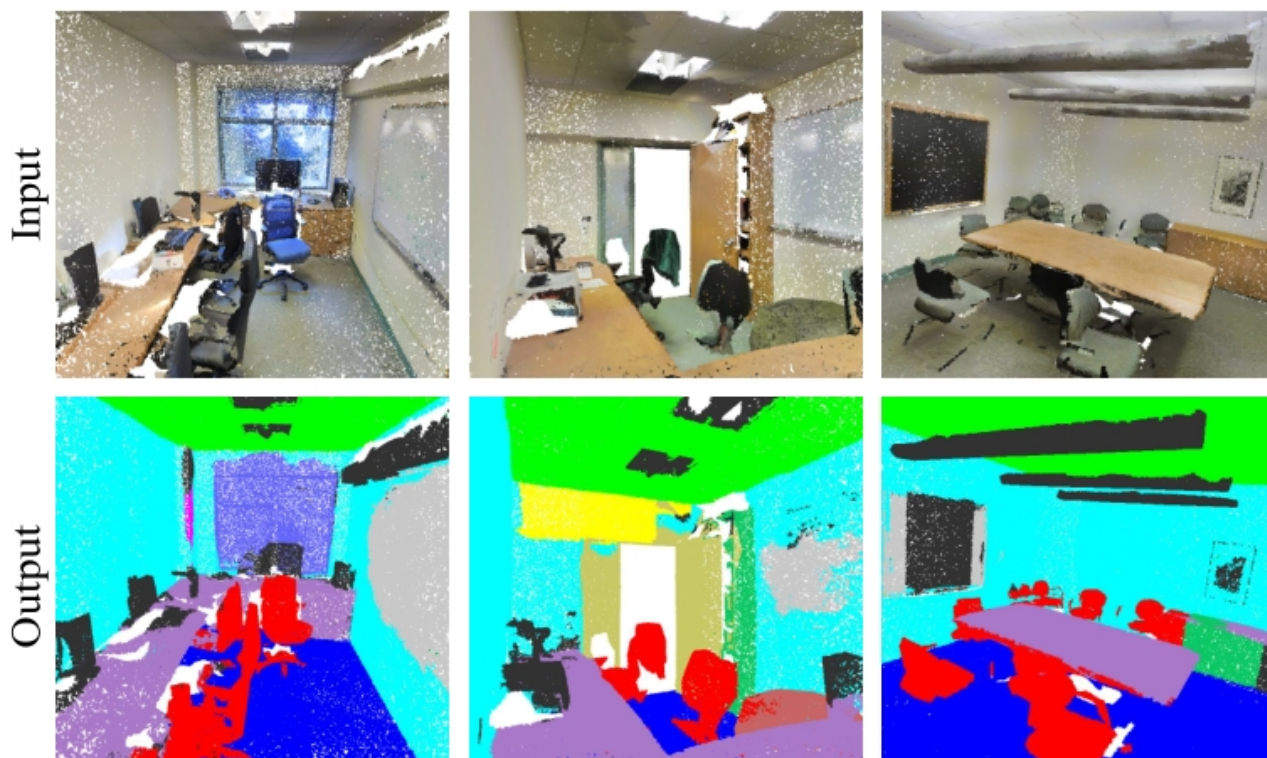
large-scale indoor spaces 相同的协议，使用 k-折叠策略进行训练和测试。

我们将我们的方法与使用手工制作的点特征的基线进行比较。该基线提取了相同的 9 维局部特征和 3 个额外的特征：局部点密度、局部曲率和法线，使用标准的 MLP 作为分类器。

实验结果

	mean IoU	overall accuracy
Ours baseline	20.12	53.19
Ours PointNet	47.71	78.62

由表中的定量结果可见，PointNet 方法的性能明显优于基线方法。



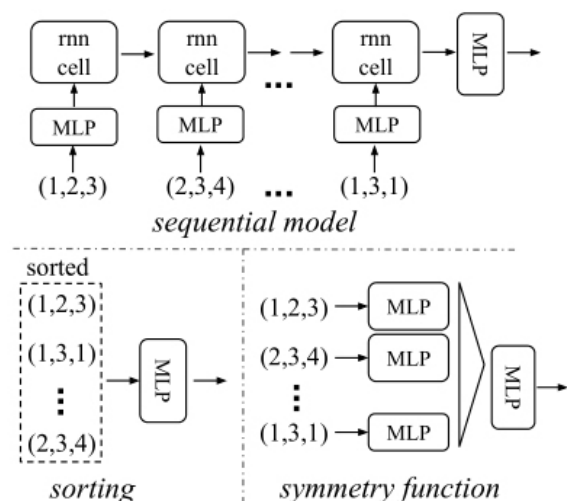
上图显示了定性的分割结果。我们的网络能够输出平滑的预测，并且对缺失点和遮挡具有很强的鲁棒性。

5.2. 架构设计分析

该部分实验通过对照实验来验证我们的设计选择，还展示了网络超参数的影响。

5.2.1 与其他顺序不变方法的比较

该对照实验仍然使用 ModelNet40 形状分类问题作为比较这些方案的试验台。



	accuracy
MLP (unsorted input)	24.2
MLP (sorted input)	45.0
LSTM	78.5
Attention sum	83.0
Average pooling	83.8
Max pooling	87.1

我们比较的基线包括 $n \times 3$ 阵列的未排序和排序的点上的多层感知机、将输入点视为序列的RNN模型和基于对称函数的模型。

我们实验的对称操作包括最大汇集、平均汇集和基于注意力的加权和。注意力方法类似于 **Order Matters** 中的方法，其中从每个点特征预测分数，然后通过计算 $\text{Softmax}^{1, 2}$ 来对分数进行跨点归一化。然后对归一化分数和点特征计算加权和。如图上所示，最大池化操作以较大的优势实现了最佳性能，这验证了我们的选择。

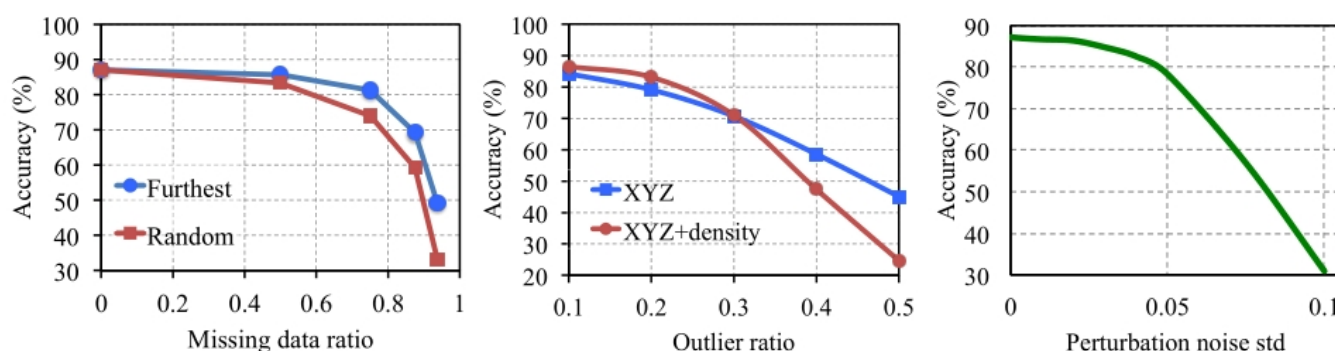
- [1] 初探softmax。
- [2] 为什么是SoftMax?

5.2.2 输入和特征转换的有效性

Transform	accuracy
none	87.1
input (3x3)	87.9
feature (64x64)	86.9
feature (64x64) + reg.	87.4
both	89.2

在上表中，我们展示了输入和特性转换(用于对齐)的积极效果。看到最基本的架构已经取得了相当合理的结果，使用输入转换可使性能提升0.8%。**正则化损失**是高维变换工作所必需的。通过**将变换和正则化项结合**，我们获得了最好的性能。

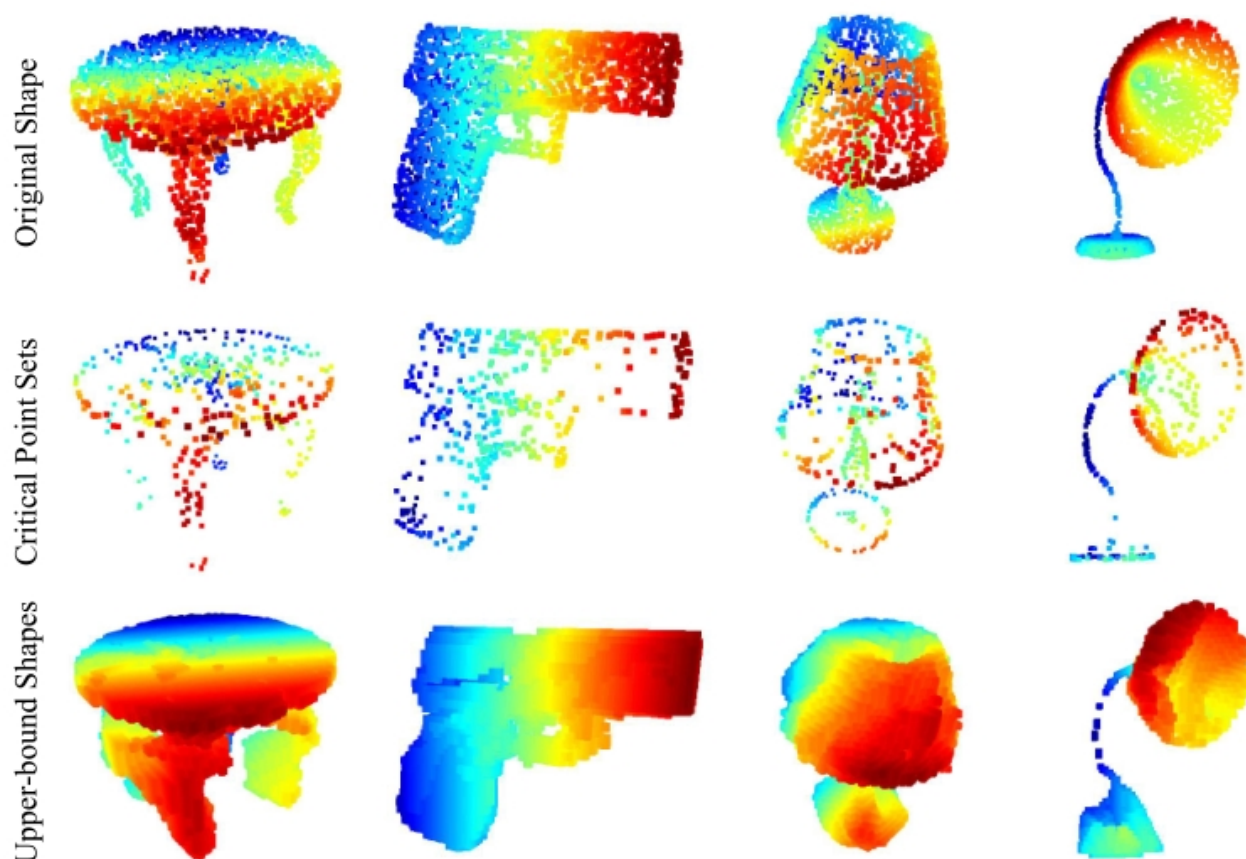
5.2.3 稳健性测试



我们使用与 5.2.1 的最大池化网络相同的架构，输入点被标准化到单位球体中，结果如上图所示。对于缺失点，当有 50% 的缺失点时，采用最远随机输入采样，准确率仅下降 2.4% 和 3.8%。我们评估了两个模型：一个针对具有 (x, y, z) 坐标的点进行训练；另一个针对 (x, y, z) 加点密度进行训练，即使当 20 的点异常时，该网络也有 80 以上的准确率。

5.3. 可视化 PointNet

虽然临界点共同确定给定形状的全局形状特征，但任何落在临界点集和上界形状之间的点云都会提供完全相同的特征。我们对所有数字进行颜色编码，以显示深度信息。



上图可视化了一些样本形状 S 的临界点集 C_S 和上界形状 N_S 。这两个形状之间的点集将给出完全相同的全局形状特征 $f(S)$ 。

临界点集 C_S 描述了形状的骨架，上界形状 N_S 表示给出与输入点云 S 相同的全局形状特征 $f(S)$ 的最大可能的点云，这意味着丢失一些非临界点根本不会改变全局形状，反映了 PointNet 的健壮性。

5.4. 时间和空间复杂度分析

	#params	FLOPs/sample
PointNet (vanilla)	0.8M	148M
PointNet	3.5M	440M
Subvolume [18]	16.6M	3633M
MVCNN [23]	60.0M	62057M

上表总结了 PointNet 的空间(网络中的参数数量)和时间(浮点运算/样本)复杂度。

我们还将 PointNet 与以前工作中一组有代表性的基于体积和多视图的体系结构进行了比较。

虽然 MVCNN 和 Subvolume(3D CNN) 实现了高性能，但 PointNet 在计算成本方面的效率更高(以 FLOP/样本衡量：效率分别提高141倍和8倍)。此外，就网络中的 $\#params$ 而言，PointNet 的空间效率比 MVCNN 高得多(参数减少了17倍)。

PointNet 的可扩展性更强，它的空间和时间复杂度在输入点数量上是 $O(N)$ 线性的。然而，由于卷积在计算时间上占主导地位，多视角方法的时间复杂度与图像分辨率成正比增长，而基于体积卷积的方法则随

体积大小呈三次曲线增长。

根据经验，使用 TensorFlow 上的1080X GPU，PointNet 能够每秒处理超过一百万个点，用于点云分类(约1K对象/秒)或语义分割(约2个房间/秒)，显示出巨大的实时应用潜力。

六、结论

在这项工作中，我们提出了一种新的直接处理点云的深度神经网络 PointNet。我们的网络为许多 3D 识别任务提供了统一的方法，包括对象分类、部件分割和语义分割，同时在标准基准上获得了与最先进水平相当或更好的结果。我们还为理解我们的网络提供理论分析和可视化。