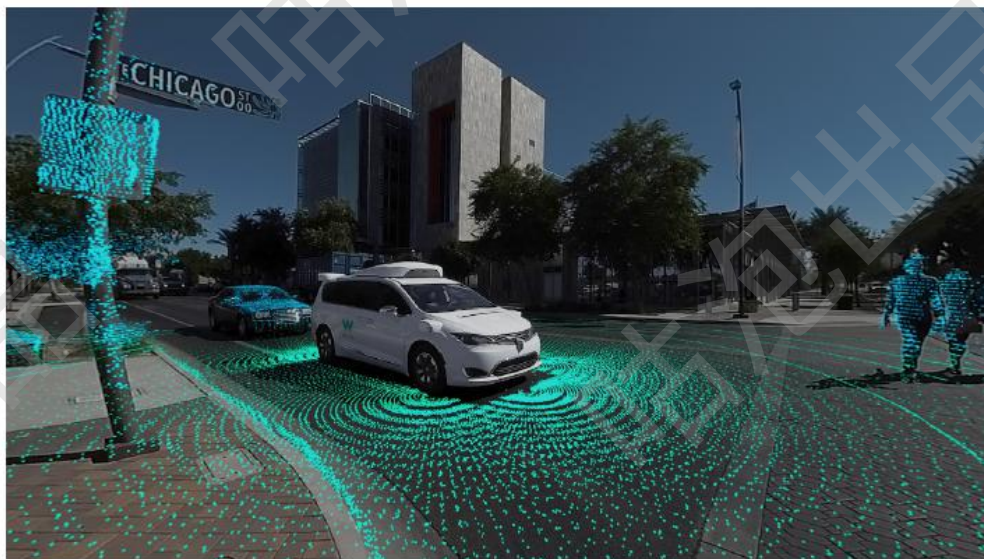


# PointNet

✓ 3D数据应用领域:

✎ backbone! backbone! backbone!

Self-Driving Cars



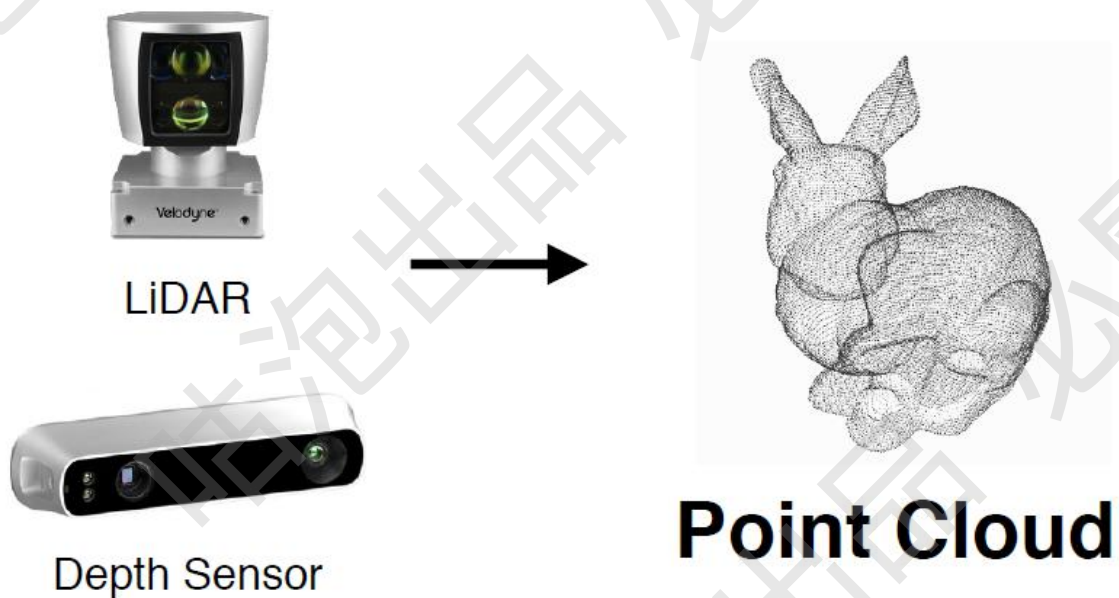
Augmented Reality



# PointNet

✓ 如何获取3D数据:

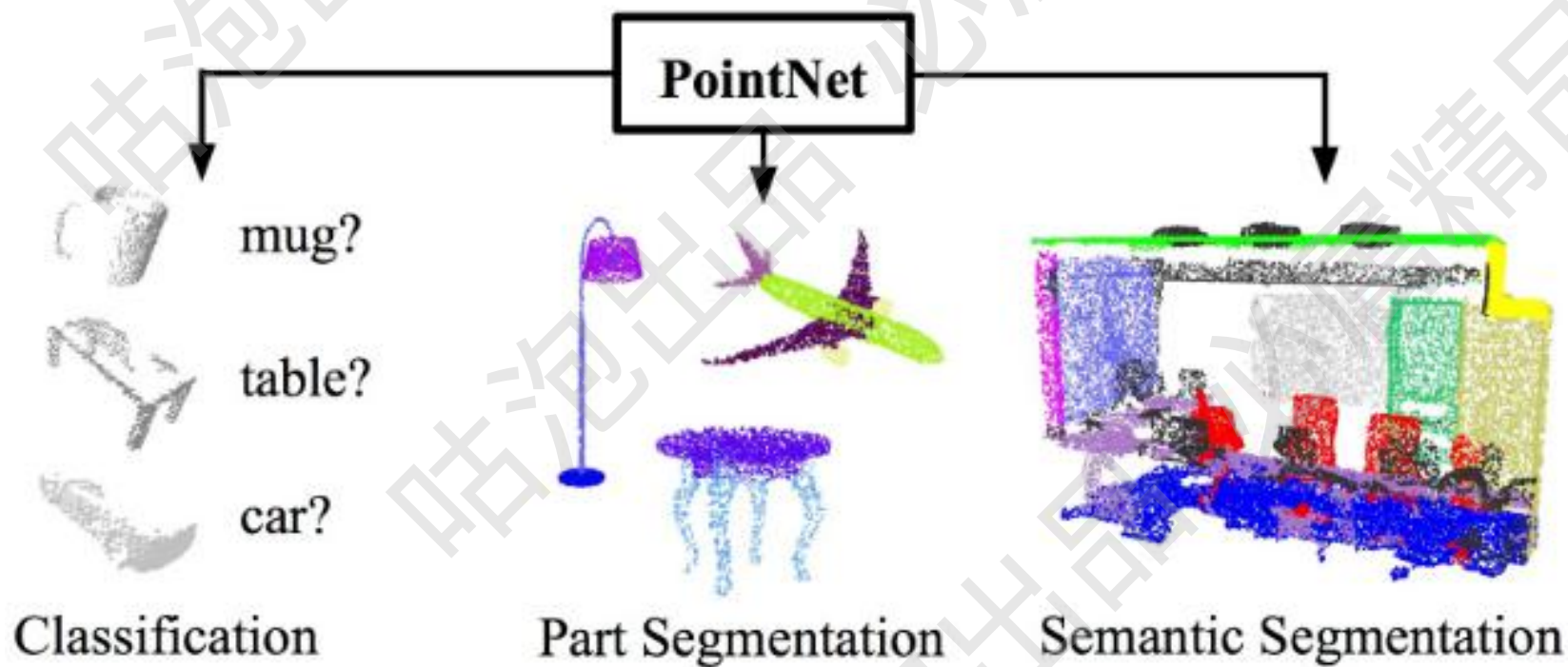
✎ 传感器扫描后得到的就是点云数据, 如何利用点云数据呢?



# PointNet

✓ 通用网络架构

✎ 分类任务；部件分割；场景分割等





# PointNet

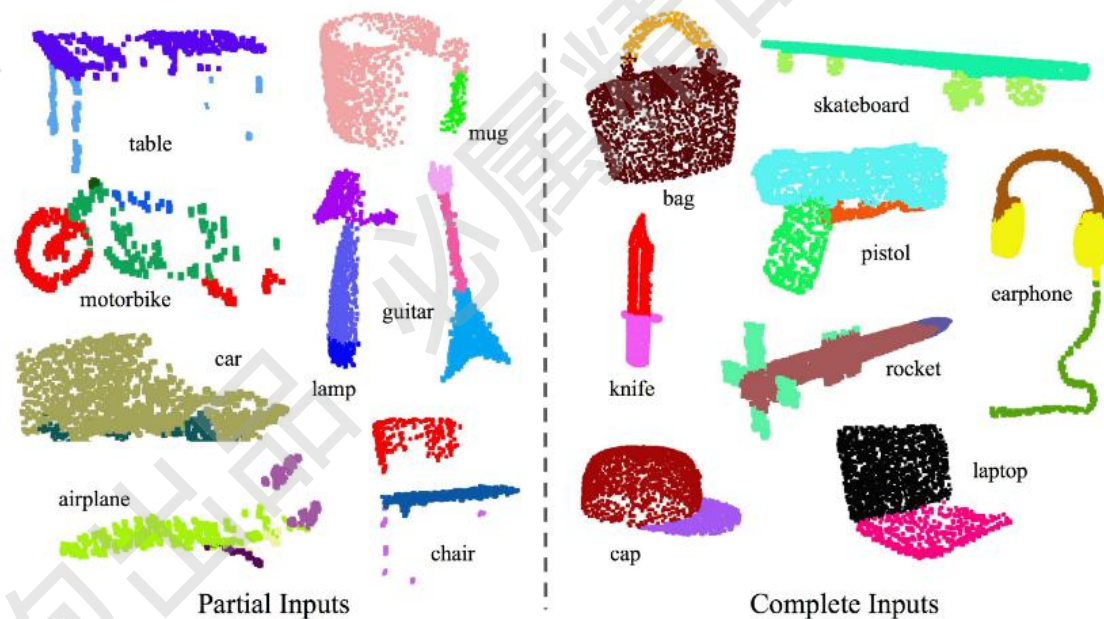
## ✓ 通用网络架构

✎ 左图：场景分割任务；右图：部件分割任务

*Semantic Segmentation Results*



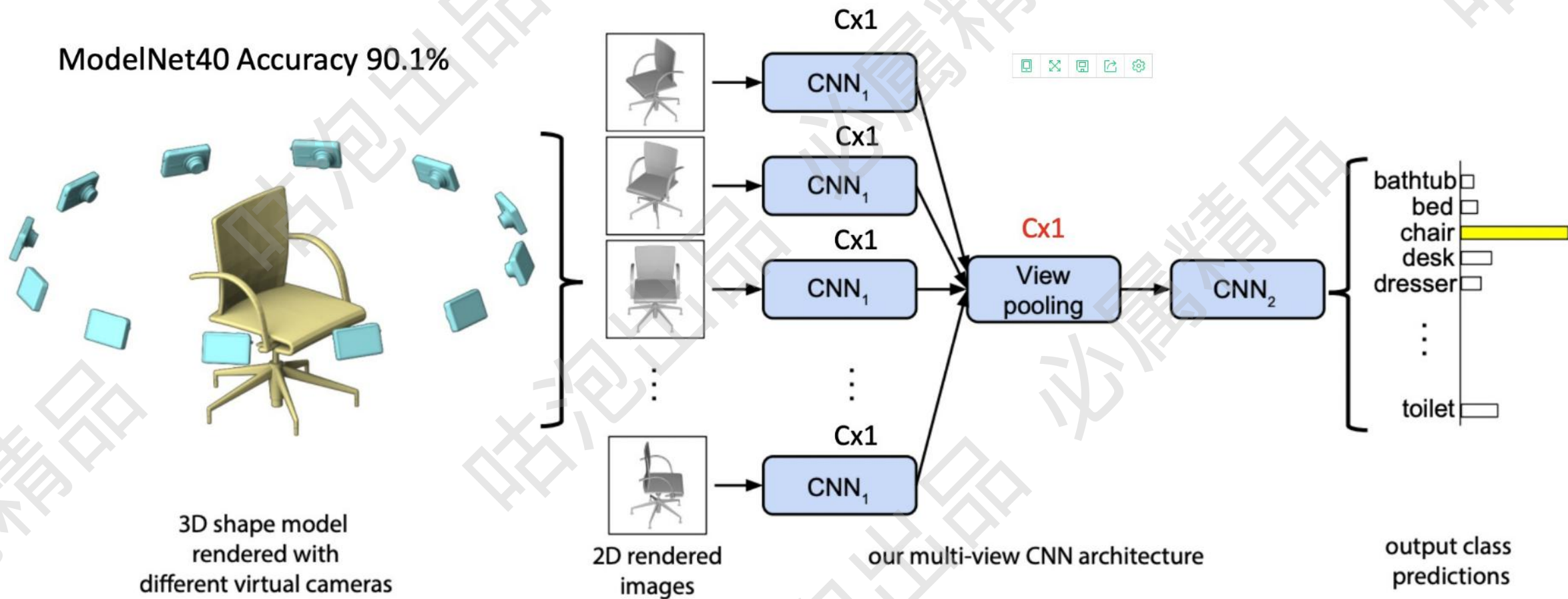
*Object Part Segmentation Results*



# PointNet

✓ 如何处理3维数据?

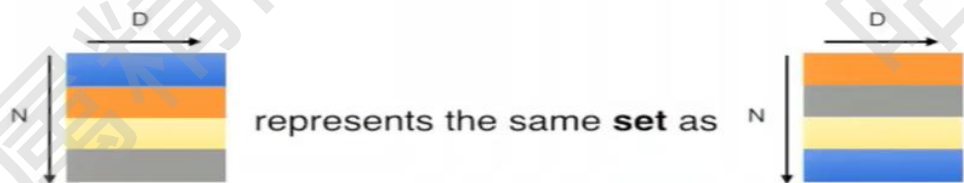
ModelNet40 Accuracy 90.1%



# PointNet

✓ 点云数据:

✎ 无序性: 只是点而已, 排列顺序不影响



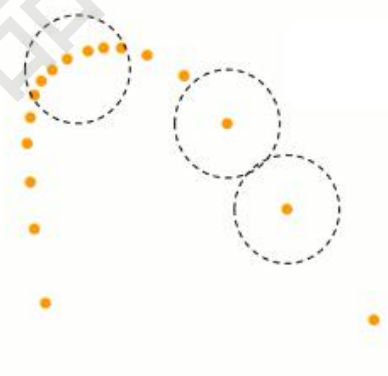
✎ 近密远疏的特性: 扫描与视角不同导致



✎ 非结构化数据, 直接CNN有点难



✎ 要解决的任务就是如何对点云数据进行特征提取



# PointNet

✓ 能不能省掉那些预处理操作而直接利用点云呢?

✎ 当下深度学习的核心思想就是一条龙服务(End2End)





# PointNet

✓ PointNet基本出发点:

✎ 由于点的无序性导致, 需要模型具有置换不变性

$$f(x_1, x_2, \dots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \dots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

✎ 例如如下公式, 但是如何在神经网络中体现出来呢?

$$f(x_1, x_2, \dots, x_n) = \max\{x_1, x_2, \dots, x_n\}$$

$$f(x_1, x_2, \dots, x_n) = x_1 + x_2 + \dots + x_n$$



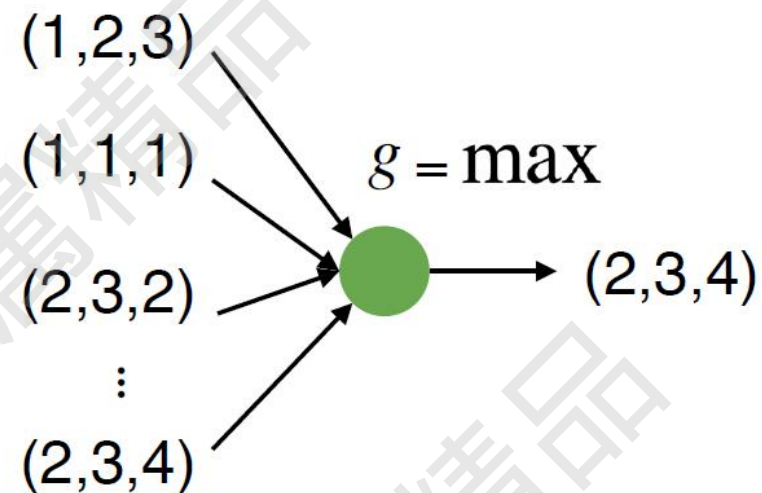
# PointNet

✓ 基本出发点:

✎ 直接利用Max函数（简单粗暴）：

✎ 又是七伤拳，损失太多特征了，怎么办？

✎ 能不能先升维然后再做Max操作（其实就是神经网络的隐层）

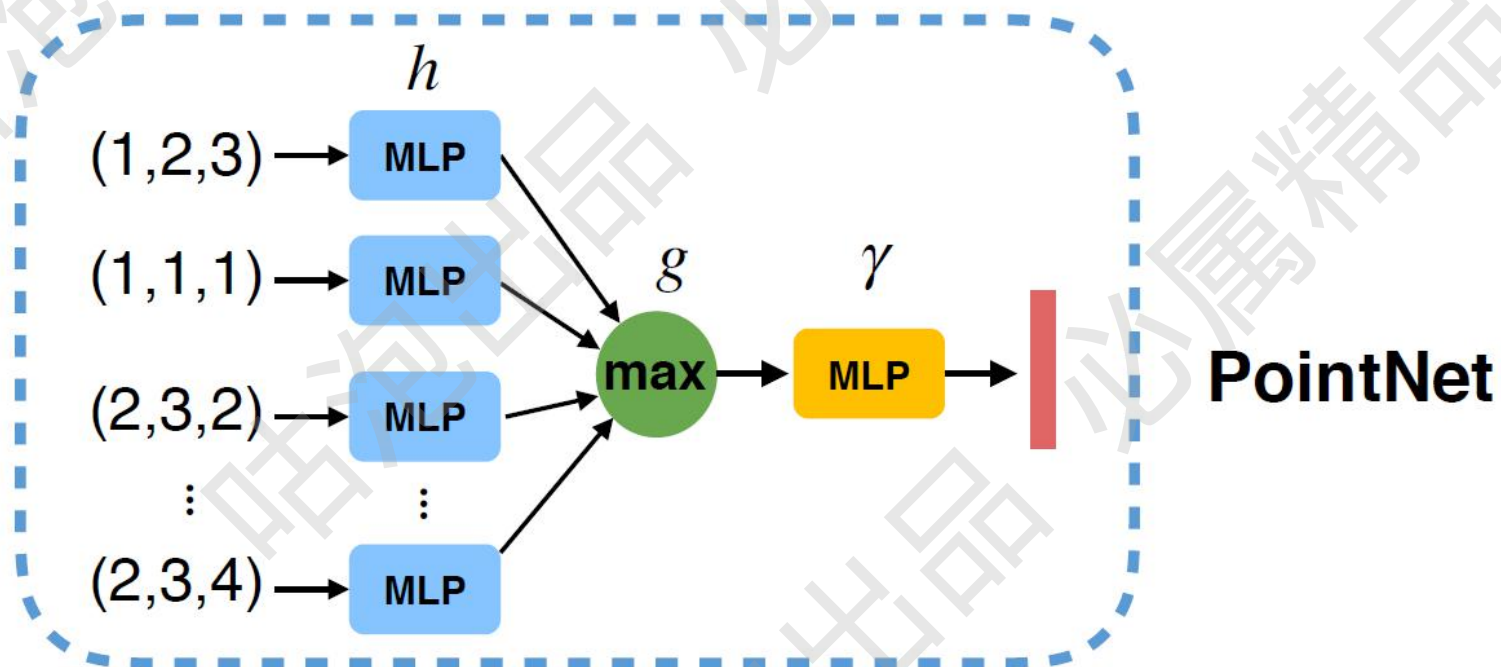


$$f(x_1, x_2, \dots, x_n) = \gamma \circ g(h(x_1), \dots, h(x_n))$$

# PointNet

✓ 基本模型架构:

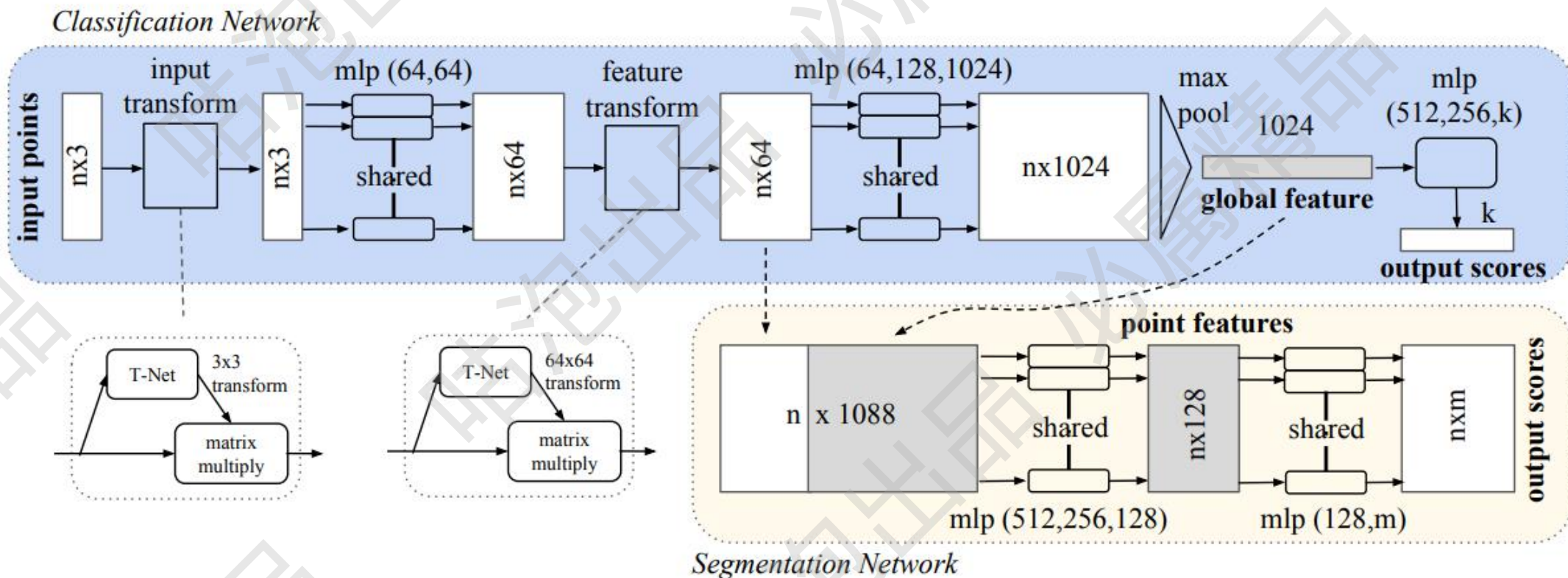
✎ 分别对每个点进行特征提取（卷积或者全连接），再MAX得到全局进行输出



# PointNet

✓ 整体网络架构:

✎ 分类就是得到整体特征再输出；分割就是各个点特征输出结果



# PointNet++

✓ PointNet有哪些问题呢？

✎ 跟当下主流网络不符，没有局部特征融合，要不自己，要么一个整体。

✎ 没有关系概念，局部样本点之间肯定存在关系的，没有考虑到

✎ PointNet++版本要从局部入手，多利用局部特征

✎ 整体思想不变，只不过在特征提取处使用类似图卷积的方式来整合特征



# PointNet++

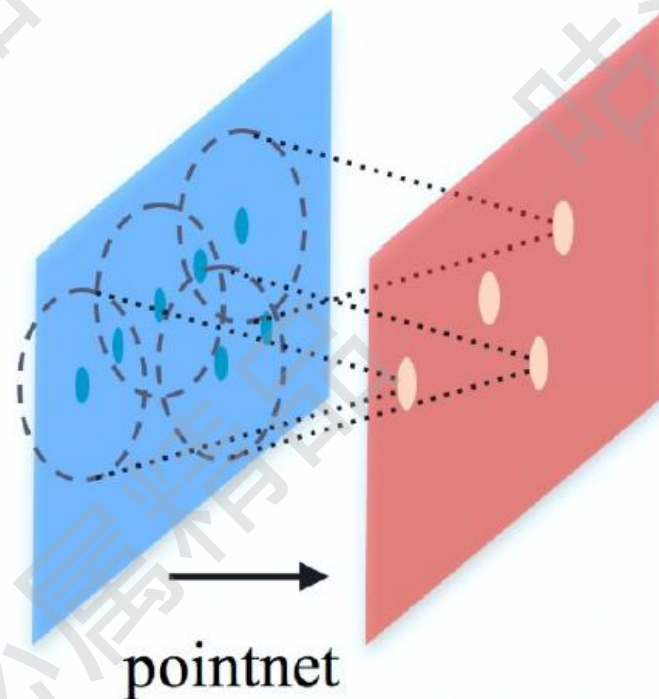
✓ 基本出发点:

✎ 基于半径选择局部区域 (类似得到很多个簇)

✎ 针对得到的每个区域进行特征提取 (卷积)

✎ 要解决的问题: 如何选择区域 (簇中心点选择)

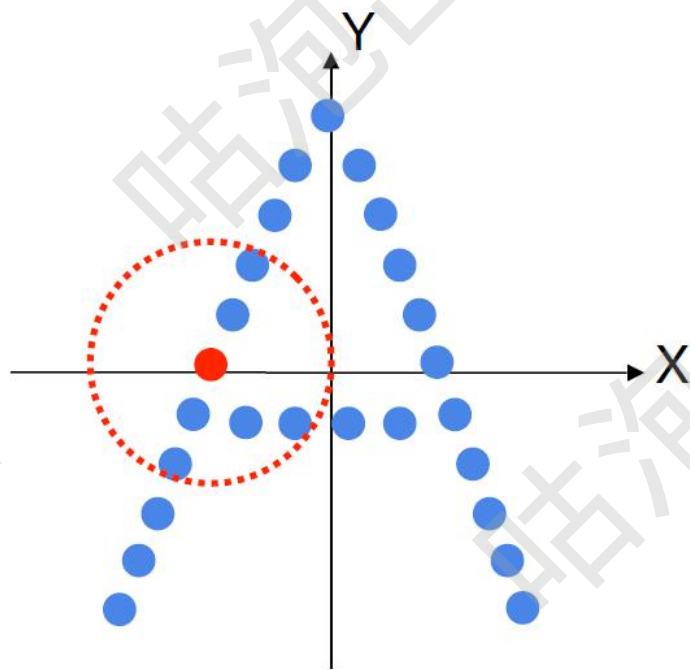
✎ 簇的半径大小如何定义, 每个簇中选择多少个样本点



# PointNet++

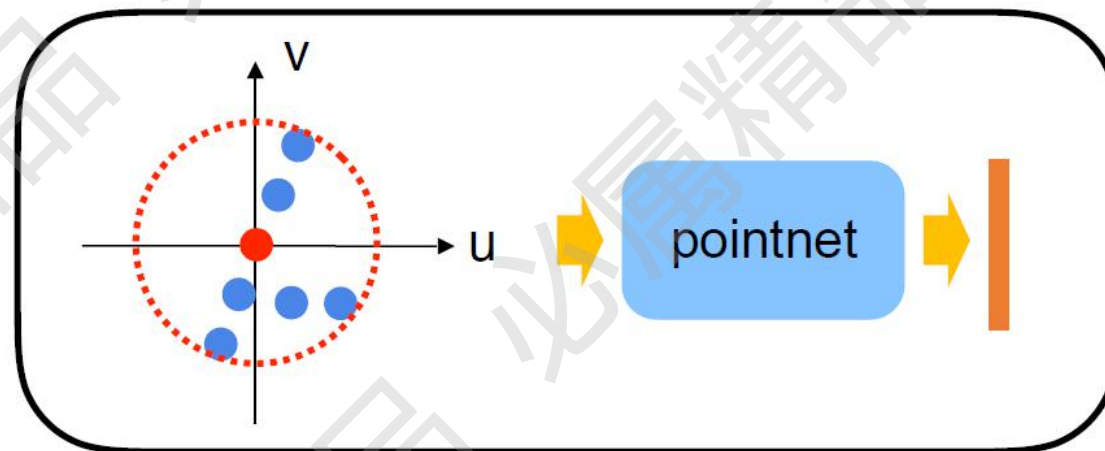
✓ 基本出发点:

✎ 第一步先确定好每一个局部区域, 接下来对局部区域执行pointnet



N points in (X,Y)

Apply pointnet at a local region

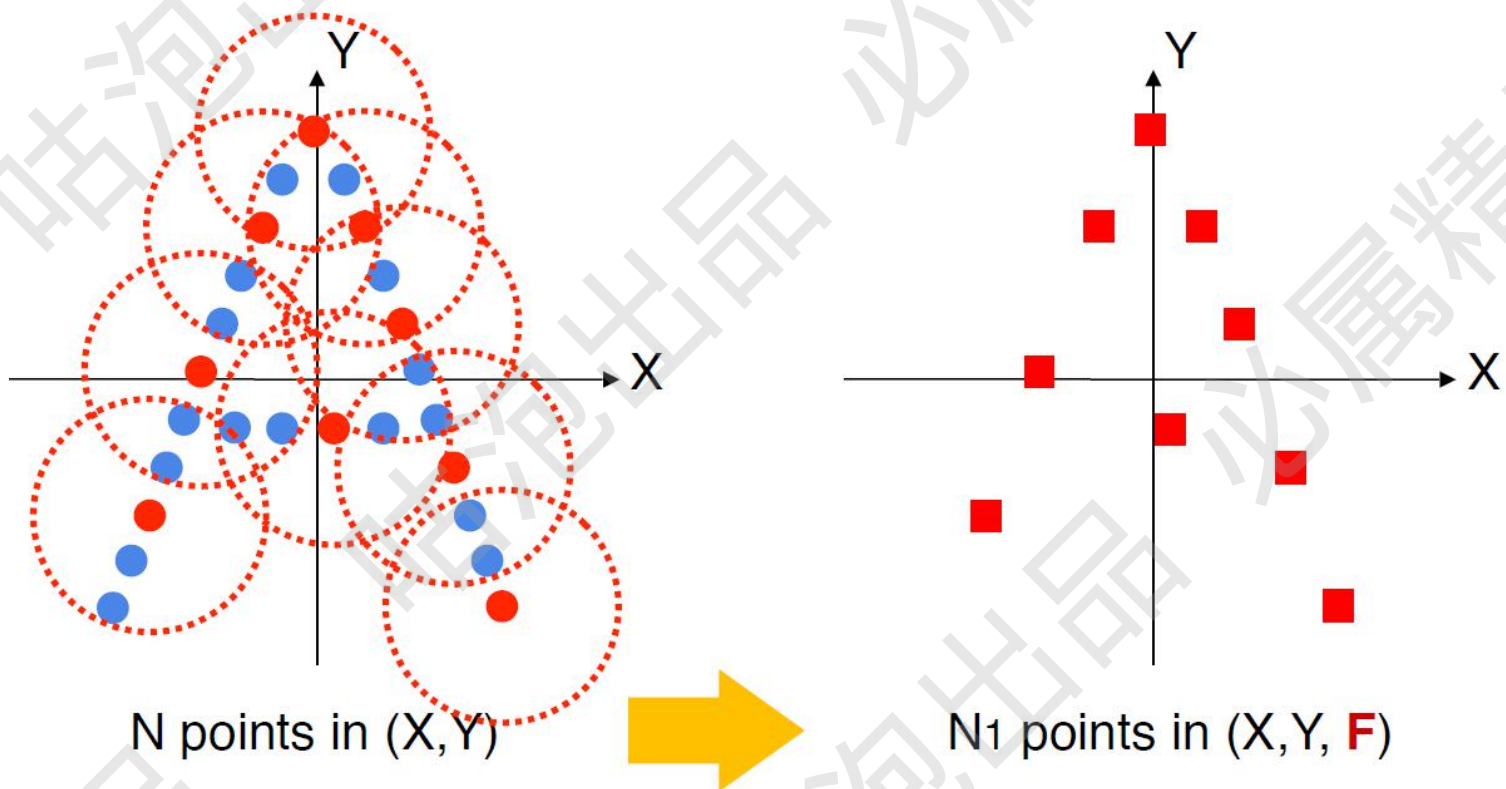


k points in local  
coordinates (u,v)

# PointNet++

✓ 最远点采样 (farthest point sampling)

✎ 例如输入1024个点，要选择128个中心点（簇），如何采样呢？



# PointNet++

## ✓ 分组 (gouping)

✎ 例如：输入为  $\text{batch} * 1024 * 6$  (1024个点, 每个点对应3个坐标3个法向量信息)

✎ 分组后输出为:  $\text{batch} * 128 * 16 * 6$  (128个中心点, 每个簇16个样本)

✎ 实际计算时是选择多种半径, 多种样本点个数, 目的是特征更丰富

✎ 例如: 半径 = (0.1, 0.2, 0.4) ; 对应簇的样本个数 (16, 32, 64)



# PointNet++

✓ 对各组进行特征提取：

✎ 先进行维度变换 ( $b * npoints * nsample * features$ ,  $8 * 128 * 16 * 6 \rightarrow 8 * 6 * 16 * 128$ )

✎ 进行卷积操作 (例如:  $in=6$ ,  $out=64$ ) 就得到提取的特征 ( $8 * 64 * 16 * 128$ )

✎ 注意当前每个簇都是16个样本点，我们要每一个簇对应一个特征

✎ 按照pointnet，做MAX操作，得到 $8 * 64 * 128$

# PointNet++

✓ 继续做多次采样，分组，卷积：

✎ 例如：采样中心点 (1024- $\rightarrow$ 512- $\rightarrow$ 128)

✎ 每一次操作时，都要进行特征拼接（无论半径为0.1,0.2,0.4；以及簇采样点个数）

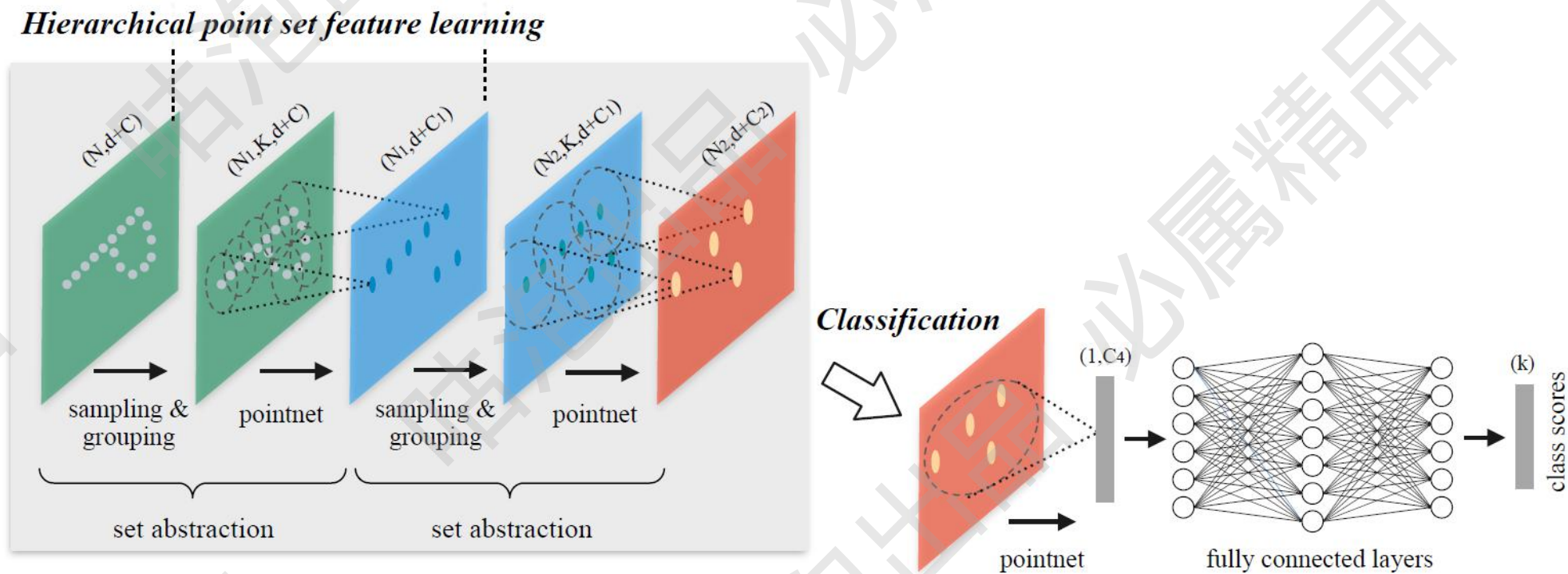
✎ 最终都得到batch\*中心点个数\*特征（但是特征个数可能不同）

✎ 执行拼接操作 ( $b*512*128, b*512*256, b*512*512$ )  $\rightarrow (b*512*896)$

# PointNet++

✓ 分类整体网络架构:

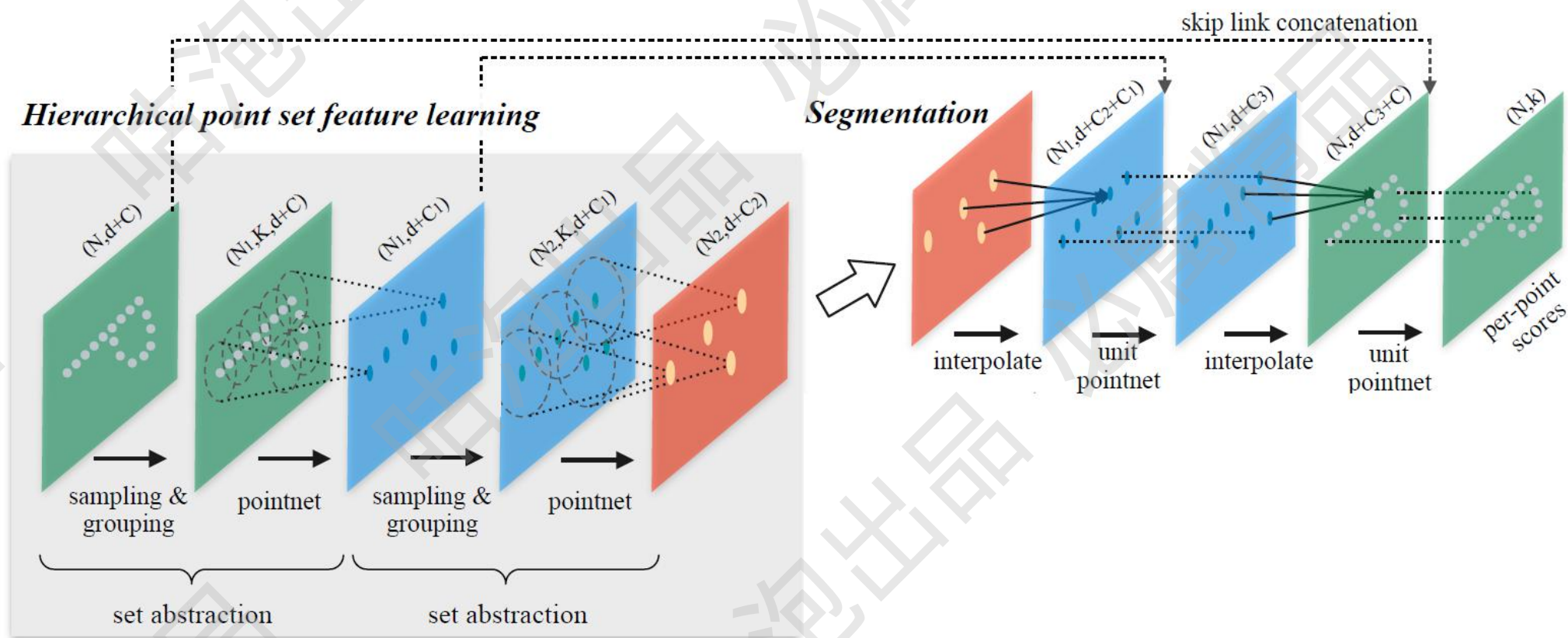
✎ 经过多次采样, 分组, pointnet得到最终整体特征, 再进行分类



# PointNet++

✓ 分割整体网络架构:

✎ 分割任务有些不同，要得到每个点的特征（还需进行上采样操作）

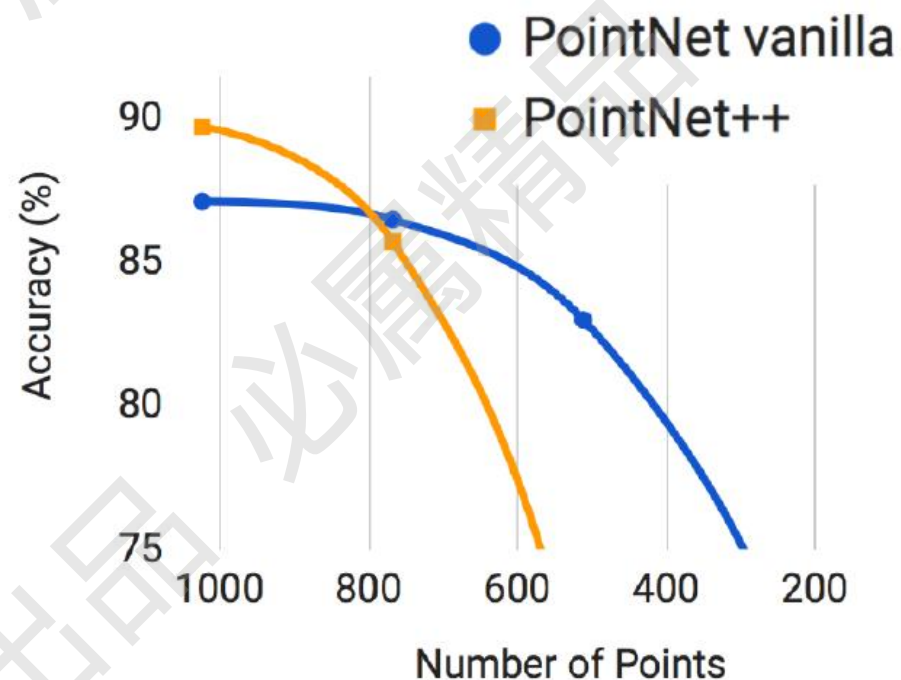
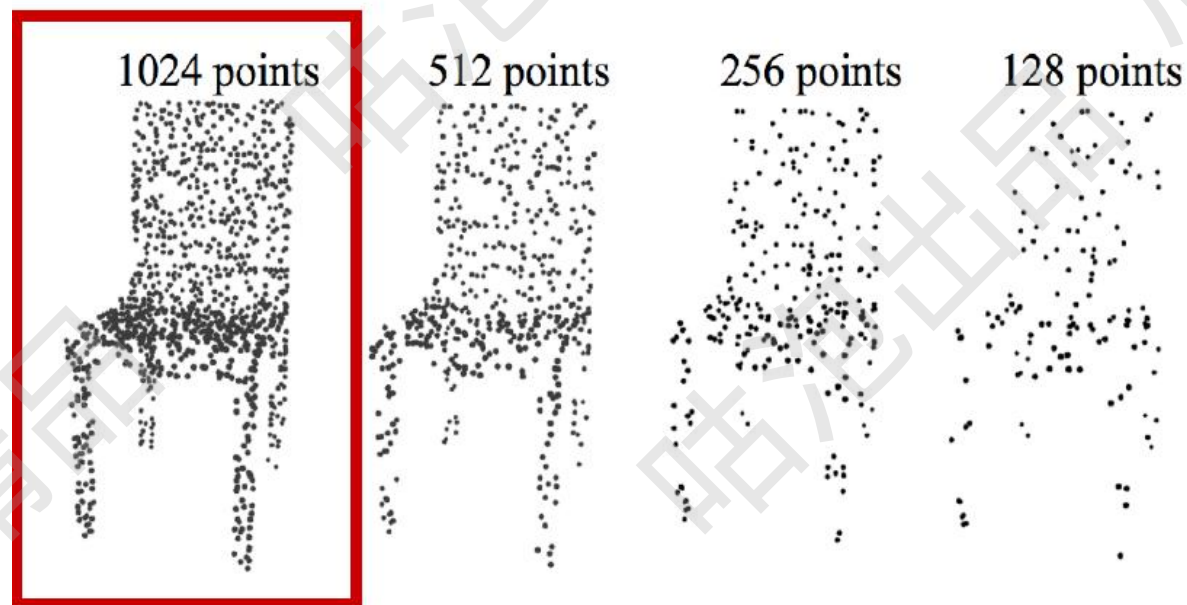




# PointNet++

✓ 遇到的问题:

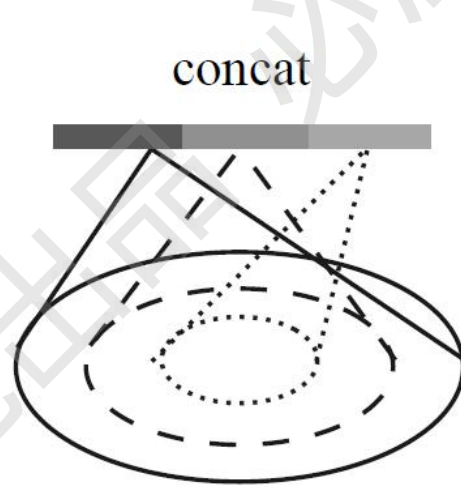
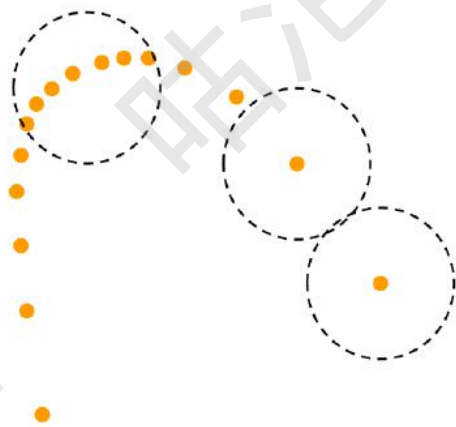
✎ PointNet++ 容易受样本点个数的影响:



# PointNet++

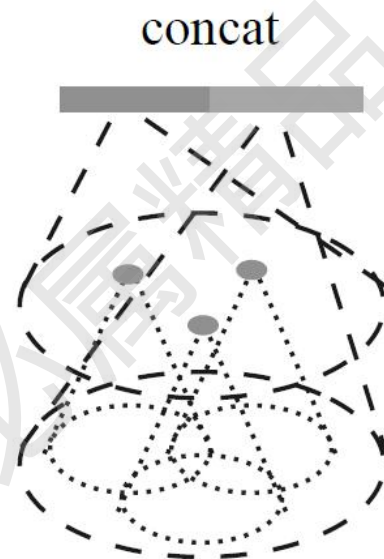
✓ 改进的设计:

✎ 多半径进行特征拼接或者跨层来提取不同分辨率特征



(a)

Multi-scale grouping (MSG)



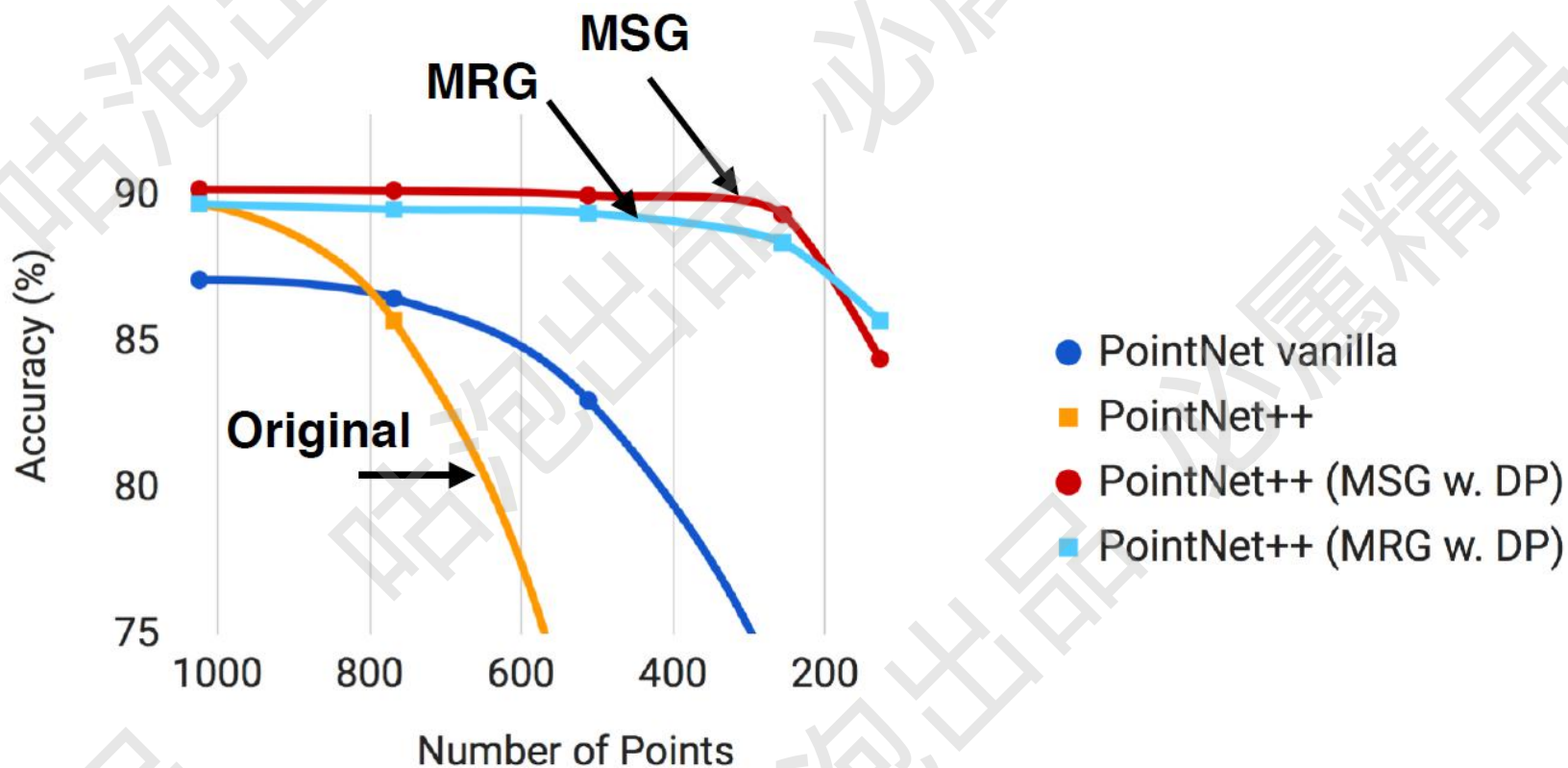
(b)

Multi-res grouping (MRG)

# PointNet++

✓ 稳定性提升:

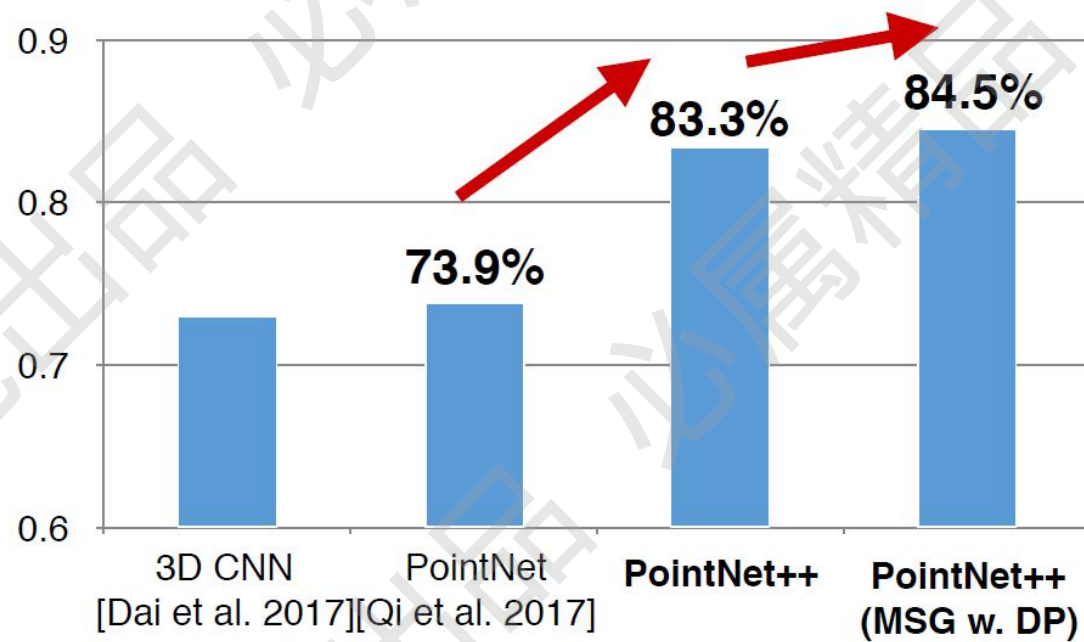
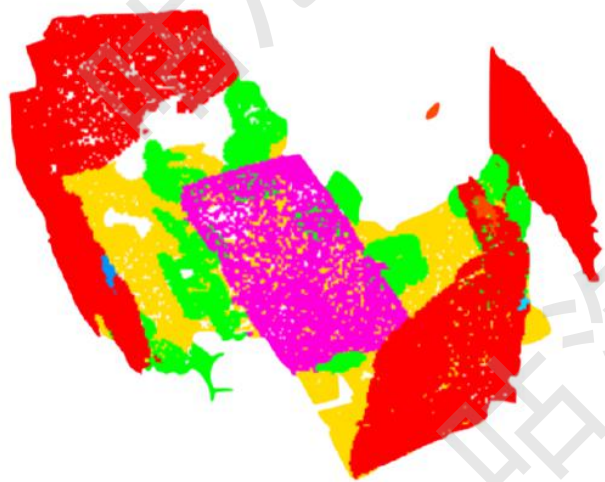
📎 采样点个数下降时候不会严重影响结果



# PointNet++

✓ 多项任务中均得到很好的效果:

✎ 主要还是应用于分类与分割任务中



*dataset: ScanNet; metric: per-point semantic classification accuracy (%)*