PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space

Charles R. Qi Li Yi Hao Su Leonidas J. Guibas
Stanford University

Abstract

Few prior works study deep learning on point sets. PointNet [20] is a pioneer in this direction. However, by design PointNet does not capture local structures induced by the metric space points live in, limiting its ability to recognize fine-grained patterns and generalizability to complex scenes. In this work, we introduce a hierarchical neural network that applies PointNet recursively on a nested partitioning of the input point set. By exploiting metric space distances, our network is able to learn local features with increasing contextual scales. With further observation that point sets are usually sampled with varying densities, which results in greatly decreased performance for networks trained on uniform densities, we propose novel set learning layers to adaptively combine features from multiple scales. Experiments show that our network called PointNet++ is able to learn deep point set features efficiently and robustly. In particular, results significantly better than state-of-the-art have been obtained on challenging benchmarks of 3D point clouds.

1 Introduction

We are interested in analyzing geometric point sets which are collections of points in a Euclidean space. A particularly important type of geometric point set is point cloud captured by 3D scanners, e.g., from appropriately equipped autonomous vehicles. As a set, such data has to be invariant to permutations of its members. In addition, the distance metric defines local neighborhoods that may exhibit different properties. For example, the density and other attributes of points may not be uniform across different locations — in 3D scanning the density variability can come from perspective effects, radial density variations, motion, etc.

Few prior works study deep learning on point sets. PointNet [20] is a pioneering effort that directly processes point sets. The basic idea of PointNet is to learn a spatial encoding of each point and then aggregate all individual point features to a global point cloud signature. By its design, PointNet does not capture local structure induced by the metric. However, exploiting local structure has proven to be important for the success of convolutional architectures. A CNN takes data defined on regular grids as the input and is able to progressively capture features at increasingly larger scales along a multi-resolution hierarchy. At lower levels neurons have smaller receptive fields whereas at higher levels they have larger receptive fields. The ability to abstract local patterns along the hierarchy allows better generalizability to unseen cases.

We introduce a hierarchical neural network, named as PointNet++, to process a set of points sampled in a metric space in a hierarchical fashion. The general idea of PointNet++ is simple. We first partition the set of points into overlapping local regions by the distance metric of the underlying space. Similar to CNNs, we extract local features capturing fine geometric structures from small neighborhoods; such local features are further grouped into larger units and processed to produce higher level features. This process is repeated until we obtain the features of the whole point set.

The design of PointNet++ has to address two issues: how to generate the partitioning of the point set, and how to abstract sets of points or local features through a local feature learner. The two issues

arXiv:1706.02413v1 [cs.CV] 7 Jun 2017

PointNet++: 在度量空间中点集上的深度分层特征学习

Charles R. Qi Li Yi Hao Su LeonidasJ.Guibas斯坦福大学

摘要

很少有先前工作研究点集上的深度学习。PointNet [20] 是这一方向上的先驱。然而，按设计，PointNet不会捕获度量空间点所生活的局部结构，这限制了它识别细粒度模式和泛化到复杂场景的能力。在这项工作中，我们引入了一个分层神经网络，它递归地应用于输入点集的嵌套划分。通过利用度量空间距离，我们的网络能够学习具有越来越大上下文尺度的局部特征。通过进一步观察到点集通常以不同的密度采样，这导致在均匀密度上训练的网络性能大大下降，我们提出了新的集学习层来自适应地组合来自多个尺度的特征。实验表明，我们称为PointNet++ 的网络能够高效且鲁棒地学习深度点集特征。特别是，在3D点云的具有挑战性的基准测试上，我们获得了显著优于当前最优的结果。

1 简介

我们对分析欧几里得空间中的点集感兴趣。几何点集的一种特别重要的类型是由 3D 扫描仪捕获的点云，例如来自适当配备的自主车辆。作为一组数据，它必须对其成员的排列保持不变。此外，距离度量定义了可能表现出不同属性局部邻域。例如，点的密度和其他属性在不同位置可能不均匀——在 3D 扫描中，密度变化可能来自透视效应、径向密度变化、运动等。

很少有先前工作研究点集上的深度学习。PointNet [20] 是一项开创性的工作，它直接处理点集。PointNet 的基本思想是学习每个点的空间编码，然后将所有单个点特征聚合成全局点云签名。根据其设计，PointNet 不会捕获由度量引起的局部结构。然而，利用局部结构已被证明对卷积架构的成功至关重要。CNN 将定义在规则网格上的数据作为输入，并且能够沿着多分辨率层次结构逐步捕获越来越大尺度的特征。在较低级别，神经元的感受野较小，而在较高级别，它们具有较大的感受野。沿着层次结构抽象局部模式的能力允许更好地泛化到未见过的案例。

我们介绍了一种层次神经网络，命名为PointNet++，以层次的方式处理在度量空间中采样的一组点。PointNet++ 的总体思路很简单。我们首先通过底层空间的距离度量将点集划分为重叠的局部区域。类似于CNN，我们从小的邻域中提取捕获精细几何结构的局部特征；这些局部特征被进一步分组到更大的单元中，并经过处理以产生更高级别的特征。这个过程重复进行，直到我们获得整个点集的特征。

PointNet++ 的设计需要解决两个问题：如何生成点集的划分，以及如何通过局部特征学习器抽象点集或局部特征。这两个问题

are correlated because the partitioning of the point set has to produce common structures across partitions, so that weights of local feature learners can be shared, as in the convolutional setting. We choose our local feature learner to be PointNet. As demonstrated in that work, PointNet is an effective architecture to process an unordered set of points for semantic feature extraction. In addition, this architecture is robust to input data corruption. As a basic building block, PointNet abstracts sets of local points or features into higher level representations. In this view, PointNet++ applies PointNet recursively on a nested partitioning of the input set.

One issue that still remains is how to generate overlapping partitioning of a point set. Each partition is defined as a neighborhood ball in the underlying Euclidean space, whose parameters include centroid location and scale. To evenly cover the whole set, the centroids are selected among input point set by a farthest point sampling (FPS) algorithm. Compared with volumetric CNNs that scan the space with fixed strides, our local receptive fields are dependent on both the input data and the metric, and thus more efficient and effective.

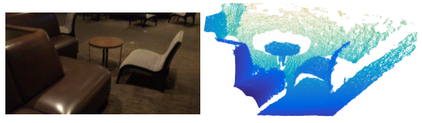


Figure 1: Visualization of a scan captured from a Structure Sensor (left: RGB; right: point cloud).

Deciding the appropriate scale of local neighborhood balls, however, is a more challenging yet intriguing problem, due to the entanglement of feature scale and non-uniformity of input point set. We assume that the input point set may have variable density at different areas, which is quite common in real data such as Structure Sensor scanning [18] (see Fig. 1). Our input point set is thus very different from CNN inputs which can be viewed as data defined on regular grids with uniform constant density. In CNNs, the counterpart to local partition scale is the size of kernels. [25] shows that using smaller kernels helps to improve the ability of CNNs. Our experiments on point set data, however, give counter evidence to this rule. Small neighborhood may consist of too few points due to sampling deficiency, which might be insufficient to allow PointNets to capture patterns robustly.

A significant contribution of our paper is that PointNet++ leverages neighborhoods at multiple scales to achieve both robustness and detail capture. Assisted with random input dropout during training, the network learns to adaptively weight patterns detected at different scales and combine multi-scale features according to the input data. Experiments show that our PointNet++ is able to process point sets efficiently and robustly. In particular, results that are significantly better than state-of-the-art have been obtained on challenging benchmarks of 3D point clouds.

2 Problem Statement

Suppose that $\mathcal{X} = (M, d)$ is a discrete metric space whose metric is inherited from a Euclidean space $\mathbb{R}^n$, where $M \subseteq \mathbb{R}^n$ is the set of points and d is the distance metric. In addition, the density of M in the ambient Euclidean space may not be uniform everywhere. We are interested in learning set functions f that take such $\mathcal{X}$ as the input (along with additional features for each point) and produce information of semantic interest regrading $\mathcal{X}$. In practice, such f can be classification function that assigns a label to $\mathcal{X}$ or a segmentation function that assigns a per point label to each member of M.

3 Method

Our work can be viewed as an extension of PointNet [20] with added hierarchical structure. We first review PointNet (Sec. 3.1) and then introduce a basic extension of PointNet with hierarchical structure (Sec. 3.2). Finally, we propose our PointNet++ that is able to robustly learn features even in non-uniformly sampled point sets (Sec. 3.3).

3.1 Review of PointNet [20]: A Universal Continuous Set Function Approximator

Given an unordered point set $\{x_1, x_2, ..., x_n\}$ with $x_i \in \mathbb{R}^d$, one can define a set function $f : \mathcal{X} \to \mathbb{R}$ that maps a set of points to a vector:

$$f(x_1, x_2, ..., x_n) = \gamma\left(\underset{i=1,...,n}{\text{MAX}} \{h(x_i)\}\right) \quad (1)$$

是相关的，因为点集的划分必须在不同划分之间产生共同结构，以便局部特征学习器的权重可以共享，就像卷积设置中那样。我们选择我们的局部特征学习器为PointNet。正如该工作中的演示所示，PointNet是一种有效的架构，用于处理无序点集进行语义特征提取。此外，这种架构对输入数据损坏具有鲁棒性。作为一个基本构建块，PointNet将局部点集或特征抽象为更高级别的表示。从这个角度来看，PointNet++ 在输入集的嵌套划分上递归地应用PointNet。

仍然存在的一个问题是如何生成点集的重叠划分。每个划分被定义为底层欧几里得空间中的一个邻域球，其参数包括质心位置和尺度。为了均匀覆盖整个集合，质心是通过最远点采样（FPS）算法从输入点集中选择的。与以固定步长扫描空间的体积CNN相比，我们的局部感受野取决于输入数据和度量，因此更高效和有效。

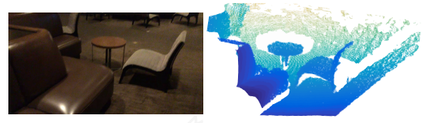


图1：Structure Sensor扫描的可视化（左：RGB；右：点云）。

然而，确定局部邻域球的适当尺度是一个更具挑战性但也引人入胜的问题，因为特征尺度和输入点集非均匀性相互交织。我们假设输入点集在不同区域可能具有可变的密度，这在Structure Sensor扫描等真实数据中很常见（见图1）。因此，我们的输入点集与CNN输入有很大不同，后者可以被视为在具有均匀恒定密度的规则网格上定义的数据。在CNN中，局部划分尺度的对应物是卷积核的大小。 [25] 表明使用较小的卷积核有助于提高CNN的能力。然而，我们对点集数据的实验给出了与此规则相反的证据。由于采样不足，小的邻域可能包含过多的点，这可能不足以让PointNets稳健地捕获模式。

我们论文的一个重要贡献是，PointNet++ 利用多尺度邻域来实现鲁棒性和细节捕获。在训练过程中辅助以随机输入dropout，网络学习自适应地加权在不同尺度上检测到的模式，并根据输入数据组合多尺度特征。实验表明，我们的PointNet++ 能够高效且鲁棒地处理点集。特别是在3D点云的挑战性基准测试中，我们获得了显著优于当前最优水平的成果。

2 问题陈述

假设 $\mathcal{X} = (M, d)$ 是一个离散度量空间，其度量继承自欧几里得空间$\mathbb{R}^n$，其中 $M \subseteq \mathbb{R}^n$ 是点的集合， d 是距离度量。此外， M在环境欧几里得空间中的密度可能并非处处均匀。我们感兴趣于学习集合函数 f，该函数以这种 $\mathcal{X}$ 作为输入（以及每个点的附加特征）并产生关于 $\mathcal{X}$的语义信息。在实践中，这种 f 可以是一个将标签分配给 $\mathcal{X}$ 的分类函数，或是一个将每个点标签分配给 M每个成员的分割函数。

3 方法

我们的工作可以看作是 PointNet [20] 的扩展，并增加了层次结构。我们首先回顾 PointNet (Sec. 3.1)，然后介绍一个具有层次结构的 PointNet 的基本扩展 (Sec. 3.2)。最后，我们提出了我们的 PointNet++，它能够在非均匀采样的点集中稳健地学习特征 (Sec. 3.3)。

3.1 PointNet [20] 回顾：一个通用的连续集函数逼近器

给定一个无序点集 $\{x_1, x_2, ..., x_n\}$，使用 $x_i \in \mathbb{R}^d$，可以定义一个集合函数 f： $\mathcal{X} \to \mathbb{R}$ 将一个点集映射到一个向量：

$$f(x_1, x_2, ..., x_n) = \gamma \quad \underset{i=1,...,n}{\text{MAX}} \{h(x_i)\} \quad (1)$$

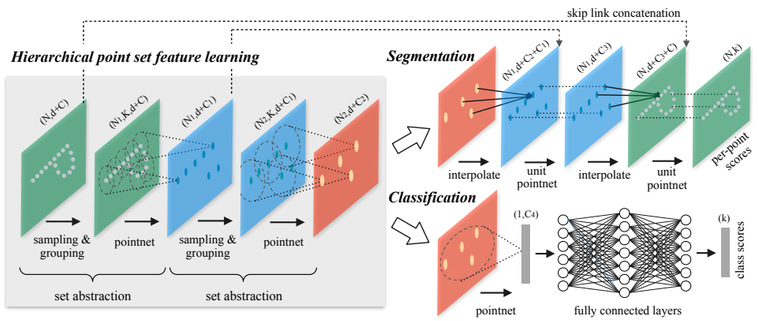


Figure 2: Illustration of our hierarchical feature learning architecture and its application for set segmentation and classification using points in 2D Euclidean space as an example. Single scale point grouping is visualized here. For details on density adaptive grouping, see Fig. 3

where γ and h are usually multi-layer perceptron (MLP) networks.

The set function f in Eq. 1 is invariant to input point permutations and can arbitrarily approximate any continuous set function [20]. Note that the response of h can be interpreted as the spatial encoding of a point (see [20] for details).

PointNet achieved impressive performance on a few benchmarks. However, it lacks the ability to capture local context at different scales. We will introduce a hierarchical feature learning framework in the next section to resolve the limitation.

3.2 Hierarchical Point Set Feature Learning

While PointNet uses a single max pooling operation to aggregate the whole point set, our new architecture builds a hierarchical grouping of points and progressively abstract larger and larger local regions along the hierarchy.

Our hierarchical structure is composed by a number of *set abstraction* levels (Fig. 2). At each level, a set of points is processed and abstracted to produce a new set with fewer elements. The set abstraction level is made of three key layers: *Sampling layer*, *Grouping layer* and *PointNet layer*. The *Sampling layer* selects a set of points from input points, which defines the centroids of local regions. *Grouping layer* then constructs local region sets by finding "neighboring" points around the centroids. *PointNet layer* uses a mini-PointNet to encode local region patterns into feature vectors.

A set abstraction level takes an $N \times (d + C)$ matrix as input that is from N points with d-dim coordinates and C-dim point feature. It outputs an $N' \times (d + C')$ matrix of N' subsampled points with d-dim coordinates and new C'-dim feature vectors summarizing local context. We introduce the layers of a set abstraction level in the following paragraphs.

Sampling layer. Given input points $\{x_1, x_2, ..., x_n\}$, we use iterative farthest point sampling (FPS) to choose a subset of points $\{x_{i_1}, x_{i_2}, ..., x_{i_m}\}$, such that x_{i_j} is the most distant point (in metric distance) from the set $\{x_{i_1}, x_{i_2}, ..., x_{i_{j-1}}\}$ with regard to the rest points. Compared with random sampling, it has better coverage of the entire point set given the same number of centroids. In contrast to CNNs that scan the vector space agnostic of data distribution, our sampling strategy generates receptive fields in a data dependent manner.

Grouping layer. The input to this layer is a point set of size $N \times (d + C)$ and the coordinates of a set of centroids of size $N' \times d$. The output are groups of point sets of size $N' \times K \times (d + C)$, where each group corresponds to a local region and K is the number of points in the neighborhood of centroid points. Note that K varies across groups but the succeeding *PointNet layer* is able to convert flexible number of points into a fixed length local region feature vector.

In convolutional neural networks, a local region of a pixel consists of pixels with array indices within certain Manhattan distance (kernel size) of the pixel. In a point set sampled from a metric space, the neighborhood of a point is defined by metric distance.

Ball query finds all points that are within a radius to the query point (an upper limit of K is set in implementation). An alternative range query is K nearest neighbor (kNN) search which finds a fixed

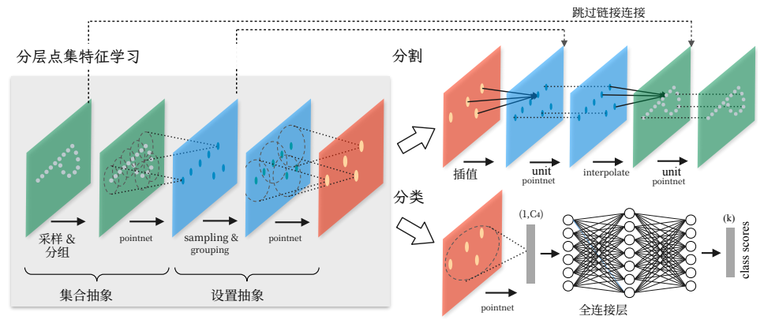


图2：我们分层特征学习架构的示意图及其在2D欧几里得空间中使用点进行集合分割和分类的应用示例。这里可视化了单尺度点分组。有关密度自适应分组的详细信息，请参见图3

其中 γ 和 h 通常是多层感知器 (MLP) 网络。

等式1中的集合函数 f 对输入点排列是不变的，并且可以任意逼近任何连续集合函数 [20]。请注意， h 的响应可以解释为点的空间编码（有关详细信息，请参见 [20] ）。

PointNet 在几个基准测试上取得了令人印象深刻的性能。然而，它缺乏在不同尺度上捕获局部上下文的能力。我们将在下一节中介绍一个分层特征学习框架来解决这个限制。

3.2 分层点集特征学习

虽然 PointNet 使用单个最大池化操作来聚合整个点集，但我们的新架构构建了点的分层分组，并沿着层次结构逐步抽象出越来越大的局部区域。

我们的分层结构由多个 集合抽象 级别组成（图 2）。在每个级别上，一组点被处理并抽象以产生一个包含更少元素的新集合。集合抽象级别由三个关键层组成：采样层、分组层 和 *PointNet* 层。 采样层 从输入点中选择一组点，这定义了局部区域的质心。 分组层 然后在质心周围找到“邻近”点来构建局部区域集。 *PointNet* 层 使用一个迷你 PointNet 将局部区域模式编码为特征向量。

一个集合抽象级别以一个来自 $N \times (d + C)$ 矩阵作为输入，该矩阵是来自 N 个具有 d-维坐标和 C-维点特征的点，并输出一个包含 N' 个子采样的点的 $N' \times (d + C')$ 矩阵，这些点的坐标为 d-维，并具有总结局部上下文的新 C'-维特征向量。我们在下一段中介绍集合抽象级别的层。

采样层。给定输入点 $\{x_1, x_2, ..., x_n\}$，我们使用迭代最远点采样（FPS）来选择点集的一个子集 $\{x_{i_1}, x_{i_2}, ..., x_{i_m}\}$，使得 x_{i_j} 是相对于其他点从集合 $\{x_{i_1}, x_{i_2}, ..., x_{i_{j-1}}\}$ 中最远的点（在度量距离上）。与随机采样相比，在相同质心数量的情况下，它对整个点集具有更好的覆盖范围。与扫描向量空间而不考虑数据分布的 CNN 相比，我们的采样策略以数据依赖的方式生成感受野。

分组层。该层的输入是一个大小为 $N \times (d + C)$ 的点集和一组大小为 $N' \times d$. 的质心坐标。输出是大小为 $N' \times K \times (d + C)$ 的点集分组，其中每个分组对应一个局部区域， K 是质心点邻域中的点数。请注意， K 在不同分组中有所不同，但随后的 *PointNet* 层 能够将可变数量的点转换为固定长度的局部区域特征向量。

在卷积神经网络中，一个像素的局部区域由与该像素在曼哈顿距离（卷积核大小）内具有数组索引的像素组成。在一个从度量空间中采样的点集中，一个点的邻域由度量距离定义。

球查询找到所有到查询点距离在半径范围内的点（实现中设置了 K 的上限）。另一种范围查询是 K 最近邻（kNN）搜索，它找到一个固定的

number of neighboring points. Compared with kNN, ball query's local neighborhood guarantees a fixed region scale thus making local region feature more generalizable across space, which is preferred for tasks requiring local pattern recognition (e.g. semantic point labeling).

PointNet layer. In this layer, the input are N' local regions of points with data size $N' \times K \times (d+C)$. Each local region in the output is abstracted by its centroid and local feature that encodes the centroid's neighborhood. Output data size is $N' \times (d + C')$.

The coordinates of points in a local region are firstly translated into a local frame relative to the centroid point: $x_i^{(j)} = x_i^{(j)} - \hat{x}^{(j)}$ for $i = 1, 2, ..., K$ and $j = 1, 2, ..., d$ where $\hat{x}$ is the coordinate of the centroid. We use PointNet [20] as described in Sec. 3.1 as the basic building block for local pattern learning. By using relative coordinates together with point features we can capture point-to-point relations in the local region.

3.3 Robust Feature Learning under Non-Uniform Sampling Density

As discussed earlier, it is common that a point set comes with non-uniform density in different areas. Such non-uniformity introduces a significant challenge for point set feature learning. Features learned in dense data may not generalize to sparsely sampled regions. Consequently, models trained for sparse point cloud may not recognize fine-grained local structures.

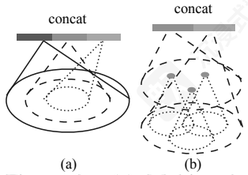


Figure 3: (a) Multi-scale grouping (MSG); (b) Multi-resolution grouping (MRG).

Ideally, we want to inspect as closely as possible into a point set to capture finest details in densely sampled regions. However, such close inspect is prohibited at low density areas because local patterns may be corrupted by the sampling deficiency. In this case, we should look for larger scale patterns in greater vicinity. To achieve this goal we propose density adaptive PointNet layers (Fig. 3) that learn to combine features from regions of different scales when the input sampling density changes. We call our hierarchical network with density adaptive PointNet layers as *PointNet++*.

Previously in Sec. 3.2, each abstraction level contains grouping and feature extraction of a single scale. In PointNet++, each abstraction level extracts multiple scales of local patterns and combine them intelligently according to local point densities. In terms of grouping local regions and combining features from different scales, we propose two types of density adaptive layers as listed below.

Multi-scale grouping (MSG). As shown in Fig. 3 (a), a simple but effective way to capture multi-scale patterns is to apply grouping layers with different scales followed by according PointNets to extract features of each scale. Features at different scales are concatenated to form a multi-scale feature.

We train the network to learn an optimized strategy to combine the multi-scale features. This is done by randomly dropping out input points with a randomized probability for each instance, which we call *random input dropout*. Specifically, for each training point set, we choose a dropout ratio θ uniformly sampled from $[0, p]$ where $p \leq 1$. For each point, we randomly drop a point with probability θ. In practice we set $p = 0.95$ to avoid generating empty point sets. In doing so we present the network with training sets of various sparsity (induced by θ) and varying uniformity (induced by randomness in dropout). During test, we keep all available points.

Multi-resolution grouping (MRG). The MSG approach above is computationally expensive since it runs local PointNet at large scale neighborhoods for every centroid point. In particular, since the number of centroid points is usually quite large at the lowest level, the time cost is significant.

Here we propose an alternative approach that avoids such expensive computation but still preserves the ability to adaptively aggregate information according to the distributional properties of points. In Fig. 3 (b), features of a region at some level L_i is a concatenation of two vectors. One vector (left in figure) is obtained by summarizing the features at each subregion from the lower level L_{i-1} using the set abstraction level. The other vector (right) is the feature that is obtained by directly processing all raw points in the local region using a single PointNet.

When the density of a local region is low, the first vector may be less reliable than the second vector, since the subregion in computing the first vector contains even sparser points and suffers more from sampling deficiency. In such a case, the second vector should be weighted higher. On the other hand,

邻近点数量。与kNN相比，球查询的局部邻域保证了固定的区域规模，从而使得局部区域特征在空间中更具泛化性，这对于需要局部模式识别的任务（例如语义点标注）更受青睐。

PointNet 层。 在这个层中，输入是 N' 点的局部区域，数据大小为 $N' \times K \times (d+C)$。输出中的每个局部区域都由其质心和编码质心邻域的局部特征进行抽象。输出数据大小为 $N' \times (d + C')$。

局部区域中点的坐标首先被转换到相对于质心点的局部坐标系： $x_i^{(j)} = x_i^{(j)} - \hat{x}^{(j)}$ 对于 $i = 1, 2, ..., K$ 和 $j = 1, 2, ..., d$，其中 $\hat{x}$ 是质心的坐标。我们使用 Sec. 3.1 中描述的 PointNet [20] 作为局部模式学习的基本构建块。通过使用相对坐标和点特征，我们可以捕获局部区域中的点对点关系。

3.3 非均匀采样密度下的鲁棒特征学习

如前所述，点集在不同区域通常具有非均匀密度。这种非均匀性给点集特征学习带来了重大挑战。在密集数据中学习到的特征可能无法泛化到稀疏采样区域。因此，为稀疏点云训练的模型可能无法识别细粒度的局部结构。

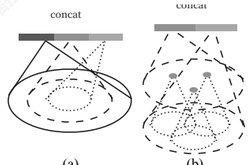


图3： (a) 多尺度分组 (MSG)； (b) 多分辨率分组 (MRG)。

理想情况下，我们希望尽可能深入地检查点集，以捕获密集采样区域中的最细微细节。然而，在低密度区域，这种近距离检查是不允许的，因为局部模式可能会因采样不足而损坏。在这种情况下，我们应该在更大的范围内寻找更大尺度的模式。为此，我们提出了密度自适应 PointNet 层 (图3)，这些层学习在输入采样密度变化时组合不同尺度区域的特征。我们将具有密度自适应 PointNet 层的层次网络称为 <style id='1'>PointNet</style>{v3}。

当输入采样密度变化时，我们称这种层次网络为*PointNet++*。

在3.2节中，每个抽象级别都包含单个尺度的分组和特征提取。在PointNet++中，每个抽象级别提取多个尺度的局部模式，并根据局部点密度智能地组合它们。在分组局部区域和组合不同尺度的特征方面，我们提出了两种类型的密度自适应层，如下所示。

多尺度分组 (MSG)。 如图3 (a)所示，捕获多尺度模式的一种简单而有效的方法是应用不同尺度的分组层，然后使用相应的PointNets提取每个尺度的特征。不同尺度的特征被连接起来形成多尺度特征。

我们训练网络学习组合多尺度特征的优化策略。这是通过随机丢弃输入点来完成的，每个实例都有随机化的概率，我们称之为随机输入丢弃。具体来说，对于每个训练点集，我们选择一个dropout比率 θ，该比率从 $[0, p]$ 中均匀采样，其中 $p \leq 1$。对于每个点，我们以概率 θ随机丢弃一个点。在实践中，我们设置 $p = 0.95$ 以避免生成空点集。通过这样做，我们向网络提供了具有各种稀疏性（由 θ诱导）和不同均匀性（由dropout中的随机性诱导）的训练集。在测试时，我们保留所有可用的点。

多分辨率分组 (MRG)。 上述 MSG 方法计算成本较高，因为它在每个质心点的大规模邻域中运行本地 PointNet。特别是，由于在最低级别质心点的数量通常很大，因此时间成本很高。

在这里，我们提出一种替代方法，它避免了如此昂贵的计算，但仍然保留了根据点的分布特性自适应聚合信息的能力。在图 3 (b) 中，某个级别 L_i 的区域特征是一个由两个向量连接而成的。一个向量（图中左侧）是通过使用集合抽象级别从较低级别的每个子区域 L_{i-1} 汇总特征获得的。另一个向量（右侧）是通过使用单个 PointNet 直接处理本地区域中所有原始点获得的特征。

当局部区域的密度较低时，第一个向量可能比第二个向量不可靠，因为计算第一个向量的子区域包含更稀疏的点，并且更多地受到采样缺陷的影响。在这种情况下，第二个向量的权重应该更高。另一方面，

when the density of a local region is high, the first vector provides information of finer details since it possesses the ability to inspect at higher resolutions recursively in lower levels.

Compared with MSG, this method is computationally more efficient since we avoids the feature extraction in large scale neighborhoods at lowest levels.

3.4 Point Feature Propagation for Set Segmentation

In set abstraction layer, the original point set is subsampled. However in set segmentation task such as semantic point labeling, we want to obtain point features for *all* the original points. One solution is to always sample all points as centroids in all set abstraction levels, which however results in high computation cost. Another way is to propagate features from subsampled points to the original points.

We adopt a hierarchical propagation strategy with distance based interpolation and across level skip links (as shown in Fig. 2). In a *feature propagation* level, we propagate point features from $N_l \times (d + C)$ points to N_{l-1} points where N_{l-1} and N_l (with $N_l \leq N_{l-1}$) are point set size of input and output of set abstraction level l. We achieve feature propagation by interpolating feature values f of N_l points at coordinates of the N_{l-1} points. Among the many choices for interpolation, we use inverse distance weighted average based on k nearest neighbors (as in Eq. 2, in default we use $p = 2, k = 3$). The interpolated features on N_{l-1} points are then concatenated with skip linked point features from the set abstraction level. Then the concatenated features are passed through a "unit pointnet", which is similar to one-by-one convolution in CNNs. A few shared fully connected and ReLU layers are applied to update each point's feature vector. The process is repeated until we have propagated features to the original set of points.

$$f^{(j)}(x) = \frac{\sum_{i=1}^{k} w_i(x) f_i^{(j)}}{\sum_{i=1}^{k} w_i(x)} \quad \text{where} \quad w_i(x) = \frac{1}{d(x, x_i)^p}, \ j = 1, ..., C \tag{2}$$

4 Experiments

Datasets We evaluate on four datasets ranging from 2D objects (MNIST [11]), 3D objects (ModelNet40 [31] rigid object, SHREC15 [12] non-rigid object) to real 3D scenes (ScanNet [5]). Object classification is evaluated by accuracy. Semantic scene labeling is evaluated by average voxel classification accuracy following [5]. We list below the experiment setting for each dataset:

- MNIST: Images of handwritten digits with 60k training and 10k testing samples.
- ModelNet40: CAD models of 40 categories (mostly man-made). We use the official split with 9,843 shapes for training and 2,468 for testing.
- SHREC15: 1200 shapes from 50 categories. Each category contains 24 shapes which are mostly organic ones with various poses such as horses, cats, etc. We use five fold cross validation to acquire classification accuracy on this dataset.
- ScanNet: 1513 scanned and reconstructed indoor scenes. We follow the experiment setting in [5] and use 1201 scenes for training, 312 scenes for test.

4.1 Point Set Classification in Euclidean Metric Space

We evaluate our network on classifying point clouds sampled from both 2D (MNIST) and 3D (ModleNet40) Euclidean spaces. MNIST images are converted to 2D point clouds of digit pixel locations. 3D point clouds are sampled from mesh surfaces from ModelNet40 shapes. In default we use 512 points for MNIST and 1024 points for ModelNet40. In last row (ours normal) in Table 2, we use face normals as additional point features, where we also use more points ($N = 5000$) to further boost performance. All point sets are normalized to be zero mean and within a unit ball. We use a three-level hierarchical network with three fully connected layers [1]

Results. In Table 1 and Table 2, we compare our method with a representative set of previous state of the arts. Note that PointNet (vanilla) in Table 2 is the the version in [20] that does not use transformation networks, which is equivalent to our hierarchical net with only one level.

Firstly, our hierarchical learning architecture achieves significantly better performance than the non-hierarchical PointNet [20]. In MNIST, we see a relative 60.8% and 34.6% error rate reduction

[1] See supplementary for more details on network architecture and experiment preparation.

当局部区域的密度较高时，第一个向量提供更精细的细节信息，因为它具有在较低级别中递归地以更高分辨率进行检查的能力。
与 MSG 相比，此方法在计算上更高效，因为我们避免了在大规模邻域最低层进行特征提取。

3.4 点特征传播用于集合分割

在集合抽象层中，原始点集会被下采样。但在集合分割任务（如语义点标注）中，我们希望为所有 原始点获取点特征。一种解决方案是在所有集合抽象层中始终将所有点作为质心进行采样，但这会导致高计算成本。另一种方法是将从下采样点传播特征到原始点。

我们采用了一种基于距离插值和跨层跳跃链接的分层传播策略（如图 2 所示）。在一个 特征传播 层中，我们将点特征从$N_l \times (d + C)$ 点传播到 N_{l-1} 点，其中 N_{l-1} 和 N_l （与 $N_l \leq N_{l-1}$）一起）是集合抽象层 l 的输入和输出点集大小。我们通过在 N_{l-1} 点的坐标处对 N_l 点的特征值f 进行插值来实现特征传播。在许多插值选择中，我们使用基于 k 个最近邻的逆距离加权平均（如 Eq. 2，默认情况下我们使用$p = 2$ $k = 3$），）。然后，将 N_{l-1} 点上的插值特征与来自集合抽象层的跳跃链接点特征连接起来。然后，将连接的特征传递通过一个“单元点网”，它类似于 CNN 中的逐个卷积。应用几个共享的全连接和 ReLU 层来更新每个点的特征向量。重复此过程，直到我们将特征传播到原始点集。

$$f^{(j)}(x) = \frac{\sum_{i=1}^{k} w_i(x) f_i^{(j)}}{\sum_{i=1}^{k} w_i(x)} \quad \text{where} \quad w_i(x) = \frac{1}{d(x, x_i)^p}, \ j = 1, ..., C \tag{2}$$

4 个实验

数据集 我们在四个数据集上评估，范围从2D对象（MNIST [11]）、3D对象（Model-Net40 [31] 刚性对象，SHREC15 [12] 非刚性对象）到真实3D场景（ScanNet [5]）。对象分类通过准确率评估。语义场景标注通过遵循 [5] 的平均体素分类准确率评估。我们列出了每个数据集的实验设置：

- MNIST：手写数字图像，有60k个训练样本和10k个测试样本。
- ModelNet40：40类（大多是人造）的CAD模型。我们使用官方分割，训练使用9,843个形状，测试使用2,468个形状。
- SHREC15：来自50个类别的1200个形状。每个类别包含24个形状，大多是各种姿势的生物形状，如马、猫等。我们使用五折交叉验证来获取该数据集的分类准确率。
- ScanNet：1513个扫描和重建的室内场景。我们遵循 [5] 中的实验设置，并使用1201个场景进行训练，312个场景进行测试。

4.1 欧几里得度量空间中的点集分类

我们在从 2D (MNIST) 和 3D (ModleNet40) 欧几里得空间中采样的点云上进行网络评估。MNIST 图像被转换为包含数字像素位置的 2D 点云。3D 点云从 ModelNet40 形状的网格表面采样。默认情况下，我们为 MNIST 使用 512 个点，为 ModelNet40 使用 1024 个点。在表 2 的最后一行（ ours normal）中，我们使用面法线作为额外的点特征，其中我们还使用更多点 ($N = 5000$) 以进一步提升性能。所有点集都被归一化，使其均值为零且位于单位球内。我们使用一个三层分层网络，具有三个全连接层 [1]

结果。 在表 1 和表 2 中，我们将我们的方法与一组具有代表性的先前最先进技术进行比较。请注意，表 2 中的 PointNet (vanilla) 是不使用变换网络的 [20] 版本，这等效于我们只有一个层级的分层网络。

首先，我们的分层学习架构在非分层 PointNet [20] 上实现了显著更好的性能。在 MNIST 上，我们观察到相对 60.8% 和 34.6% 的错误率降低

[1] 请参见补充材料以获取有关网络架构和实验准备更详细的信息。

Method	Error rate (%)
Multi-layer perceptron [24]	1.60
LeNet5 [11]	0.80
Network in Network [13]	**0.47**
PointNet (vanilla) [20]	1.30
PointNet [20]	0.78
Ours	0.51

Table 1: MNIST digit classification.

Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9**

Table 2: ModelNet40 shape classification.

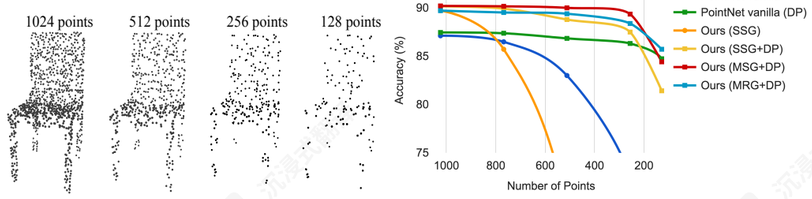


Figure 4: Left: Point cloud with random point dropout. Right: Curve showing advantage of our density adaptive strategy in dealing with non-uniform density. DP means random input dropout during training; otherwise training is on uniformly dense points. See Sec.3.3 for details.

from PointNet (vanilla) and PointNet to our method. In ModelNet40 classification, we also see that using same input data size (1024 points) and features (coordinates only), ours is remarkably stronger than PointNet. Secondly, we observe that point set based method can even achieve better or similar performance as mature image CNNs. In MNIST, our method (based on 2D point set) is achieving an accuracy close to the Network in Network CNN. In ModelNet40, ours with normal information significantly outperforms previous state-of-the-art method MVCNN [26].

Robustness to Sampling Density Variation. Sensor data directly captured from real world usually suffers from severe irregular sampling issues (Fig. 1). Our approach selects point neighborhood of multiple scales and learns to balance the descriptiveness and robustness by properly weighting them.

We randomly drop points (see Fig. 4 left) during test time to validate our network's robustness to non-uniform and sparse data. In Fig. 4 right, we see MSG+DP (multi-scale grouping with random input dropout during training) and MRG+DP (multi-resolution grouping with random input dropout during training) are very robust to sampling density variation. MSG+DP performance drops by less than 1% from 1024 to 256 test points. Moreover, it achieves the best performance on almost all sampling densities compared with alternatives. PointNet vanilla [20] is fairly robust under density variation due to its focus on global abstraction rather than fine details. However loss of details also makes it less powerful compared to our approach. SSG (ablated PointNet++ with single scale grouping in each level) fails to generalize to sparse sampling density while SSG+DP amends the problem by randomly dropping out points in training time.

4.2 Point Set Segmentation for Semantic Scene Labeling

To validate that our approach is suitable for large scale point cloud analysis, we also evaluate on semantic scene labeling task. The goal is to predict semantic object label for points in indoor scans. [5] provides a baseline using fully convolutional neural network on voxelized scans. They purely rely on scanning geometry instead of RGB information and report the accuracy on a per-voxel basis. To make a fair comparison, we remove RGB information in all our experiments and convert point cloud label prediction into voxel labeling following [5]. We also compare with [20]. The accuracy is reported on a per-voxel basis in Fig. 5 (blue bar).

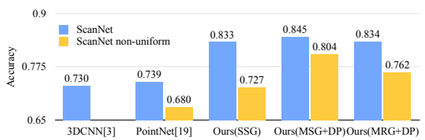


Figure 5: Scannet labeling accuracy.

Our approach outperforms all the baseline methods by a large margin. In comparison with [5], which learns on voxelized scans, we directly learn on point clouds to avoid additional quantization error,

方法	错误率 (%)
多层感知器 [24]	1.60
LeNet5 [11]	0.80
Network in Network [13]	**0.47**
PointNet (vanilla) [20]	1.30
PointNet [20]	0.78
Ours	0.51

表1：MNIST数字分类。

方法	输入	准确率 (%)
子卷积 [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (原版) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
我们 (含正常)	pc	**91.9**

表2：ModelNet40形状分类。

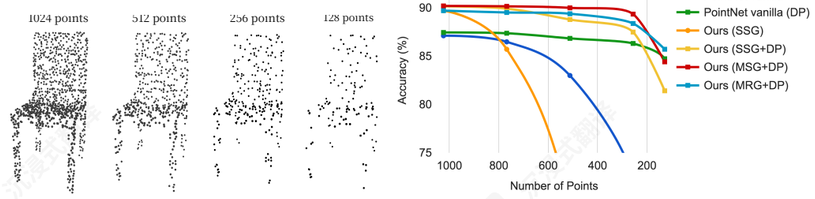


图4：左：随机点丢弃的点云。右：曲线显示我们的密度自适应策略在处理非均匀密度时的优势。DP表示训练期间随机输入丢弃；否则在均匀密度的点上训练。见第3.3节详细信息。

从 PointNet (vanilla) 到我们的方法。在 ModelNet40 分类中，我们也看到使用相同的输入数据大小（1024 个点）和特征（仅坐标），我们的方法明显强于 PointNet。其次，我们观察到基于点集的方法甚至可以实现比成熟的图像 CNN 更好或相似的性能。在 MNIST 中，我们的方法（基于 2D 点集）实现了接近 Network in Network CNN 的准确率。在 ModelNet40 中，我们的方法（带有正常信息）显著优于之前的最先进方法 MVCNN [26]。

对采样密度变化的鲁棒性。传感器数据直接从现实世界捕获通常遭受严重的非均匀采样问题（图1）。我们的方法选择多尺度点邻域并学习通过适当加权它们来平衡描述性和鲁棒性。

我们在测试期间随机丢弃点（见图4左）以验证我们网络的非均匀和稀疏数据的鲁棒性。在图4右，我们看到MSG+DP（多尺度分组，训练期间随机输入丢弃）和MRG+DP（多分辨率分组，训练期间随机输入丢弃）对采样密度变化非常鲁棒。MSG+DP从1024到256个测试点的性能下降不到1%。此外，与其他方法相比，它在几乎所有采样密度上实现了最佳性能。PointNet vanilla [20] 由于其专注于全局抽象而不是细节，在密度变化下相当鲁棒。然而，细节的丢失也使其与我们的方法相比不那么强大。SSG（在每个级别中具有单尺度分组的移除PointNet++）无法泛化到稀疏采样密度，而SSG+DP通过在训练时间随机丢弃点来修正这个问题。

4.2 点云分割用于语义场景标注

为了验证我们的方法适用于大规模点云分析，我们还评估了语义场景标注任务。目标是预测室内扫描中点的语义对象标签。 [5]使用全卷积神经网络在体素化扫描上提供了一个基线。他们纯粹依赖于扫描几何形状，而不是RGB信息，并报告了基于每个体素的精度。为了进行公平的比较，

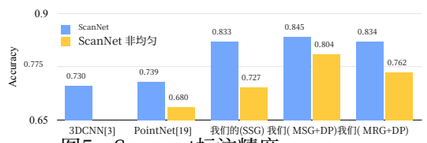


图5：Scannet标注精度。

我们在所有实验中移除了RGB信息，并按照 [5]将点云标签预测转换为体素标签。我们还与 [20]进行了比较。精度在图5（蓝条）中基于每个体素报告。

我们的方法在所有基线方法上都取得了显著的优越性。与在体素化扫描上学习的 [5]，相比，我们直接在点云上进行学习以避免额外的量化误差，

and conduct data dependent sampling to allow more effective learning. Compared with [20], our approach introduces hierarchical feature learning and captures geometry features at different scales. This is very important for understanding scenes at multiple levels and labeling objects with various sizes. We visualize example scene labeling results in Fig. 6.

Robustness to Sampling Density Variation To test how our trained model performs on scans with non-uniform sampling density, we synthesize virtual scans of Scannet scenes similar to that in Fig. 1 and evaluate our network on this data. We refer readers to supplementary material for how we generate the virtual scans. We evaluate our framework in three settings (SSG, MSG+DP, MRG+DP) and compare with a baseline approach [20].

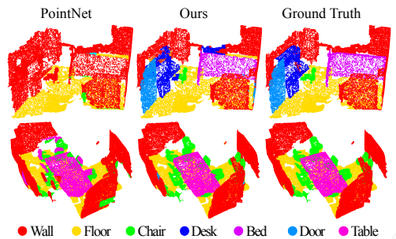


Figure 6: Scannet labeling results. [20] captures the overall layout of the room correctly but fails to discover the furniture. Our approach, in contrast, is much better at segmenting objects besides the room layout.

Performance comparison is shown in Fig. 5 (yellow bar). We see that SSG performance greatly falls due to the sampling density shift from uniform point cloud to virtually scanned scenes. MRG network, on the other hand, is more robust to the sampling density shift since it is able to automatically switch to features depicting coarser granularity when the sampling is sparse. Even though there is a domain gap between training data (uniform points with random dropout) and scanned data with non-uniform density, our MSG network is only slightly affected and achieves the best accuracy among methods in comparison. These prove the effectiveness of our density adaptive layer design.

4.3 Point Set Classification in Non-Euclidean Metric Space

In this section, we show generalizability of our approach to non-Euclidean space. In non-rigid shape classification (Fig. 7), a good classifier should be able to classify (a) and (c) in Fig. 7 correctly as the same category even given their difference in pose, which requires knowledge of intrinsic structure. Shapes in SHREC15 are 2D surfaces embedded in 3D space. Geodesic distances along the surfaces naturally induce a metric space. We show through experiments that adopting PointNet++ in this metric space is an effective way to capture intrinsic structure of the underlying point set.

For each shape in [12], we firstly construct the metric space induced by pairwise geodesic distances. We follow [23] to obtain an embedding metric that mimics geodesic distance. Next we extract intrinsic point features in this metric space including WKS [1], HKS [27] and multi-scale Gaussian curvature [16]. We use these features as input and then sample and group points according to the underlying metric space. In this way, our network learns to capture multi-scale intrinsic structure that is not influenced by the specific pose of a shape. Alternative design choices include using XYZ coordinates as points feature or use Euclidean space $\mathbb{R}^3$ as the underlying metric space. We show below these are not optimal choices.


(a) Horse (b) Cat (c) Horse
Figure 7: An example of non-rigid shape classification.

Results. We compare our methods with previous state-of-the-art method [14] in Table 3. [14] extracts geodesic moments as shape features and use a stacked sparse autoencoder to digest these features to predict shape category. Our approach using non-Euclidean metric space and intrinsic features achieves the best performance in all settings and outperforms [14] by a large margin.

Comparing the first and second setting of our approach, we see intrinsic features are very important for non-rigid shape classification. XYZ feature fails to reveal intrinsic structures and is greatly influenced by pose variation. Comparing the second and third setting of our approach, we see using geodesic neighborhood is beneficial compared with Euclidean neighborhood. Euclidean neighborhood might include points far away on surfaces and this neighborhood could change dramatically when shape affords non-rigid deformation. This introduces difficulty for effective weight sharing since the local structure could become combinatorially complicated. Geodesic neighborhood on surfaces, on the other hand, gets rid of this issue and improves the learning effectiveness.

并进行数据相关的采样以实现更有效的学习。与 [20]，相比，我们的方法引入了层次化特征学习，并捕获了不同尺度的几何特征。这对于在多个层次上理解场景和标注不同大小的物体非常重要。我们在图 6 中可视化了示例场景标注结果。

对采样密度变化的鲁棒性 为了测试我们训练的模型在非均匀采样密度的扫描上的表现，我们合成与图1相似的Scannet场景的虚拟扫描，并在这些数据上评估我们的网络。我们建议读者参考补充材料了解我们如何生成虚拟扫描。我们在三种设置（SSG、MSG+DP、MRG+DP）中评估我们的框架，并与基线方法 [20] 进行比较。

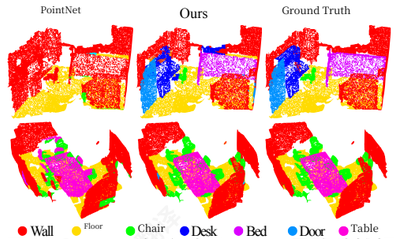


图6：Scannet标注结果。 [20]正确地捕捉了房间的整体布局，但未能发现家具。相比之下，我们的方法在分割房间布局以外的物体方面表现更好。

性能比较如图5（黄色条）所示。我们看到由于从均匀点云到虚拟扫描场景的采样密度变化，SSG性能大幅下降。另一方面，MRG网络对采样密度变化更鲁棒，因为它能够在采样稀疏时自动切换到描述更粗糙粒度特征。尽管训练数据（均匀点云带随机dropout）和具有非均匀密度的扫描数据之间存在领域差距，我们的MSG网络仅受轻微影响，并在比较的方法中取得了最高精度。这证明了我们密度自适应层设计的有效性。

4.3 非欧几里得度量空间中的点集分类

在本节中，我们展示了我们的方法在非欧几里得空间中的泛化能力。在非刚性形状分类（图7）中，一个好的分类器应该能够在图7中的(a)和(c)即使给定它们在姿态上的差异，也能正确地将其分类为同一类别，这需要了解内在结构。SHREC15中的形状是嵌入在3D空间中的2D表面。沿表面的测地距离自然地诱导出一个度量空间。我们通过实验表明，在这个度量空间中采用 PointNet++ 是一种有效的方法来捕获底层点集的内在结构。

对于 [12], 中的每个形状，我们首先通过成对测地距离构建诱导的度量空间。我们遵循 [23]来获得一个模拟测地距离的嵌入度量。接下来我们在该度量空间中提取内在点特征，包括WKS [1],HKS [27]和多尺度高斯曲率 [16]。我们使用这些特征作为输入，然后根据底层度量空间对点进行采样和分组。通过这种方式，我们的网络学习捕获不受形状特定姿态影响的 多尺度内在结构。替代设计方案包括使用XYZ 坐标作为点特征或使用欧几里得空间 $\mathbb{R}^3$ 作为底层度量空间。我们下面展示了这些不是最佳选择。


(a) Horse (b) Cat (c) Horse
图7：一个非-刚性形状分类的示例。

结果。 我们在表3中将我们的方法与先前最先进的方法 [14] 进行比较。 [14] 提取测地瞬态作为形状特征，并使用堆叠稀疏自动编码器来消化这些特征以预测形状类别。我们的方法使用非欧几里得度量空间和固有特征在所有设置中均达到最佳性能，并大幅优于 [14] 。

比较我们方法的第一种和第二种设置，我们发现内在特征对于非刚性形状分类非常重要。XYZ特征无法揭示内在结构，并且受到姿态变化的影响很大。比较我们方法的第二种和第三种设置，我们发现与欧几里得邻域相比，使用测地邻域是有益的。欧几里得邻域可能包含表面上距离很远的点，并且当形状允许非刚性变形时，这种邻域可能会发生剧烈变化。这给有效的权重共享带来了困难，因为局部结构可能会变得组合复杂。另一方面，表面上的测地邻域解决了这个问题，并提高了学习效率。

	Metric space	Input feature	Accuracy (%)
DeepGM [14]	-	Intrinsic features	93.03
Ours	Euclidean	XYZ	60.18
	Euclidean	Intrinsic features	94.49
	Non-Euclidean	Intrinsic features	**96.09**

Table 3: SHREC15 Non-rigid shape classification.

4.4 Feature Visualization.

In Fig. 8 we visualize what has been learned by the first level kernels of our hierarchical network. We created a voxel grid in space and aggregate local point sets that activate certain neurons the most in grid cells (highest 100 examples are used). Grid cells with high votes are kept and converted back to 3D point clouds, which represents the pattern that neuron recognizes. Since the model is trained on ModelNet40 which is mostly consisted of furniture, we see structures of planes, double planes, lines, corners etc. in the visualization.

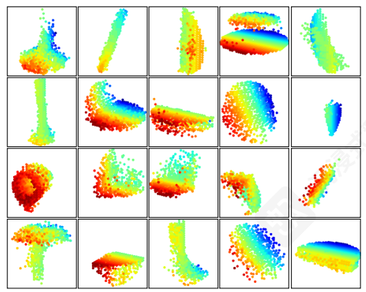


Figure 8: 3D point cloud patterns learned from the first layer kernels. The model is trained for ModelNet40 shape classification (20 out of the 128 kernels are randomly selected). Color indicates point depth (red is near, blue is far).

5 Related Work

The idea of hierarchical feature learning has been very successful. Among all the learning models, convolutional neural network [10, 25, 8] is one of the most prominent ones. However, convolution does not apply to unordered point sets with distance metrics, which is the focus of our work.

A few very recent works [20, 28] have studied how to apply deep learning to unordered sets. They ignore the underlying distance metric even if the point set does possess one. As a result, they are unable to capture local context of points and are sensitive to global set translation and normalization. In this work, we target at points sampled from a metric space and tackle these issues by explicitly considering the underlying distance metric in our design.

Point sampled from a metric space are usually noisy and with non-uniform sampling density. This affects effective point feature extraction and causes difficulty for learning. One of the key issue is to select proper scale for point feature design. Previously several approaches have been developed regarding this [19, 17, 2, 6, 7, 30] either in geometry processing community or photogrammetry and remote sensing community. In contrast to all these works, our approach learns to extract point features and balance multiple feature scales in an end-to-end fashion.

In 3D metric space, other than point set, there are several popular representations for deep learning, including volumetric grids [21, 22, 29], and geometric graphs [3, 15, 33]. However, in none of these works, the problem of non-uniform sampling density has been explicitly considered.

6 Conclusion

In this work, we propose PointNet++, a powerful neural network architecture for processing point sets sampled in a metric space. PointNet++ recursively functions on a nested partitioning of the input point set, and is effective in learning hierarchical features with respect to the distance metric. To handle the non uniform point sampling issue, we propose two novel set abstraction layers that intelligently aggregate multi-scale information according to local point densities. These contributions enable us to achieve state-of-the-art performance on challenging benchmarks of 3D point clouds.

In the future, it's worthwhile thinking how to accelerate inference speed of our proposed network especially for MSG and MRG layers by sharing more computation in each local regions. It's also interesting to find applications in higher dimensional metric spaces where CNN based method would be computationally unfeasible while our method can scale well.

	度量空间	输入特征	准确率 (%)
DeepGM [14]	-	内在特征	93.03
Ours	欧几里得	XYZ	60.18
	欧几里得	固有特征	94.49
	非欧几里得	固有特征	**96.09**

表3：SHREC15非刚性形状分类。

4.4 特征可视化。

在图8中，我们可视化了我们的分层网络的第一级核所学习的内容。我们在空间中创建了一个体素网格，并聚合在网格单元中激活某些神经元最多的局部点集（使用了最高的100个示例）。具有高票数的网格单元被保留并转换回3D点云，这代表了神经元识别的模式。由于该模型在主要由家具组成的ModelNet40上训练，我们在可视化中看到平面、双平面、线、角等结构。

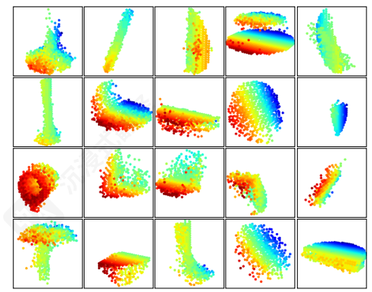


图8：从第一层核学习到的3D点云模式。模型用于ModelNet40形状分类（随机选择了128个核中的20个）。颜色表示点深度（红色近，蓝色远）。

5 相关工作

层级特征学习的思想非常成功。在所有学习模型中，卷积神经网络 [10, 25, 8] 是最突出的之一。然而，卷积不适用于具有距离度量的无序点集，这正是我们工作的重点。

A few ver y 最近的工作 [20, 28]研究了如何将深度学习应用于无序集合。即使点集确实具有距离度量，他们也忽略了潜在的度量。因此，它们无法捕获点的局部上下文，并且对全局集合的平移和归一化很敏感。在这项工作中，我们针对从度量空间采样的点，并通过在设计中明确考虑潜在的度量来解决这些问题。

从度量空间采样的点通常是嘈杂的，并且具有非均匀的采样密度。这影响了有效的点特征提取，并给学习带来了困难。一个关键问题是选择适当的点特征设计的尺度。以前在几何处理社区或摄影测量和遥感社区已经开发了几种方法来处理这个问题 [19, 17, 2, 6, 7, 30] 。与所有这些工作相比，我们的方法以端到端的方式学习提取点特征并平衡多个特征尺度。

在3D度量空间中，除了点集之外，深度学习还有几种流行的表示方法，包括体素网格 [21, 22, 29], 和几何图 [3, 15, 33]。然而，在这些工作中，非均匀采样密度的问题都没有被明确考虑。

6 结论

在这项工作中，我们提出了PointNet++，一种强大的神经网络架构，用于处理度量空间中采样的点集。PointNet++ 递归地作用于输入点集的嵌套划分，并在学习与距离度量相关的层次特征方面非常有效。为了处理非均匀点采样问题，我们提出了两个新的集合抽象层，这些层根据局部点密度智能地聚合多尺度信息。这些贡献使我们能够在具有挑战性的3D点云基准测试上实现最先进的性能。

在未来，思考如何通过在每个局部区域共享更多计算来加速我们提出的网络的推理速度，特别是对于MSG和MRG层，是值得的。同时，在更高维度的度量空间中寻找应用也很有趣，在这些空间中，基于CNN的方法在计算上不可行，而我们的方法可以很好地扩展。

References

[1] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011.

[2] D. Belton and D. D. Lichti. Classification and segmentation of terrestrial laser scanner point clouds using local variance information. *Iaprs, Xxxvi*, 5:44–49, 2006.

[3] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[4] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], 2015.

[5] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *arXiv preprint arXiv:1702.04405*, 2017.

[6] J. Demantké, C. Mallet, N. David, and B. Vallet. Dimensionality based scale selection in 3d lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 5):W12, 2011.

[7] A. Gressin, C. Mallet, J. Demantké, and N. David. Towards 3d lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS journal of photogrammetry and remote sensing*, 79:240–251, 2013.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Z. Lian, J. Zhang, S. Choi, H. ElNaghy, J. El-Sana, T. Furuya, A. Giachetti, R. A. Guler, L. Lai, C. Li, H. Li, F. A. Limberger, R. Martin, R. U. Nakanishi, A. P. Neto, L. G. Nonato, R. Ohbuchi, K. Pevzner, D. Pickup, P. Rosin, A. Sharf, L. Sun, X. Sun, S. Tari, G. Unal, and R. C. Wilson. Non-rigid 3D Shape Retrieval. In I. Pratikakis, M. Spagnuolo, T. Theoharis, L. V. Gool, and R. Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2015.

[13] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[14] L. Luciano and A. B. Hamza. Deep learning with geodesic moments for 3d shape classification. *Pattern Recognition Letters*, 2017.

[15] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 37–45, 2015.

[16] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, et al. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and mathematics*, 3(2):52–58, 2002.

[17] N. J. MITRA, A. NGUYEN, and L. GUIBAS. Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry & Applications*, 14(04n05):261–276, 2004.

[18] I. Occipital. Structure sensor-3d scanning, augmented reality, and more for mobile devices, 2016.

[19] M. Pauly, L. P. Kobbelt, and M. Gross. Point-based multiscale surface representation. *ACM Transactions on Graphics (TOG)*, 25(2):177–193, 2006.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[21] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[22] G. Riegler, A. O. Ulusoys, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. *arXiv preprint arXiv:1611.05009*, 2016.

[23] R. M. Rustamov, Y. Lipman, and T. Funkhouser. Interior distance using barycentric coordinates. In *Computer Graphics Forum*, volume 28, pages 1279–1288. Wiley Online Library, 2009.

[24] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV, to appear*, 2015.

[27] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[28] O. Vinyals, S. Bengio, and M. Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

参考文献

[1] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011.

[2] D. Belton and D. D. Lichti. Classification and segmentation of terrestrial laser scanner point clouds using local variance information. *Iaprs, Xxxvi*, 5:44–49, 2006.

[3] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[4] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], 2015.

[5] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *arXiv preprint arXiv:1702.04405*, 2017.

[6] J. Demantké, C. Mallet, N. David, and B. Vallet. Dimensionality based scale selection in 3d lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 5):W12, 2011.

[7] A. Gressin, C. Mallet, J. Demantké, and N. David. Towards 3d lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS journal of photogrammetry and remote sensing*, 79:240–251, 2013.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Z. Lian, J. Zhang, S. Choi, H. ElNaghy, J. El-Sana, T. Furuya, A. Giachetti, R. A. Guler, L. Lai, C. Li, H. Li, F. A. Limberger, R. Martin, R. U. Nakanishi, A. P. Neto, L. G. Nonato, R. Ohbuchi, K. Pevzner, D. Pickup, P. Rosin, A. Sharf, L. Sun, X. Sun, S. Tari, G. Unal, and R. C. Wilson. Non-rigid 3D Shape Retrieval. In I. Pratikakis, M. Spagnuolo, T. Theoharis, L. V. Gool, and R. Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2015.

[13] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[14] L. Luciano and A. B. Hamza. Deep learning with geodesic moments for 3d shape classification. *Pattern Recognition Letters*, 2017.

[15] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 37–45, 2015.

[16] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, et al. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and mathematics*, 3(2):52–58, 2002.

[17] N. J. MITRA, A. NGUYEN, and L. GUIBAS. Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry & Applications*, 14(04n05):261–276, 2004.

[18] I. Occipital. Structure sensor-3d scanning, augmented reality, and more for mobile devices, 2016.

[19] M. Pauly, L. P. Kobbelt, and M. Gross. Point-based multiscale surface representation. *ACM Transactions on Graphics (TOG)*, 25(2):177–193, 2006.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[21] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[22] G. Riegler, A. O. Ulusoys, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. *arXiv preprint arXiv:1611.05009*, 2016.

[23] R. M. Rustamov, Y. Lipman, and T. Funkhouser. Interior distance using barycentric coordinates. In *Computer Graphics Forum*, volume 28, pages 1279–1288. Wiley Online Library, 2009.

[24] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV, to appear*, 2015.

[27] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[28] O. Vinyals, S. Bengio, and M. Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[29] P.-S. WANG, Y. LIU, Y.-X. GUO, C.-Y. SUN, and X. TONG. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. 2017.

[30] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304, 2015.

[31] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[32] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016.

[33] L. Yi, H. Su, X. Guo, and L. Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. *arXiv preprint arXiv:1612.00606*, 2016.

[29] P.-S. WANG, Y. LIU, Y.-X. GUO, C.-Y. SUN, and X. TONG. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. 2017.

[30] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304, 2015.

[31] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[32] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016.

[33] L. Yi, H. Su, X. Guo, and L. Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. *arXiv preprint arXiv:1612.00606*, 2016.

10

Supplementary

A Overview

This supplementary material provides more details on experiments in the main paper and includes more experiments to validate and analyze our proposed method.

In Sec B we provide specific network architectures used for experiments in the main paper and also describe details in data preparation and training. In Sec C we show more experimental results including benchmark performance on part segmentation and analysis on neighborhood query, sensitivity to sampling randomness and time space complexity.

B Details in Experiments

Architecture protocol. We use following notations to describe our network architecture.

SA(K,r,[$l_1, ..., l_d$]) is a set abstraction (SA) level with K local regions of ball radius r using PointNet of d fully connected layers with width l_i ($i = 1, ..., d$). SA([$l_1, ...l_d$]) is a global set abstraction level that converts set to a single vector. In multi-scale setting (as in MSG), we use SA(K, [$r^{(1)}, ..., r^{(m)}$], [[$l_1^{(1)}, ..., l_d^{(1)}$],...,[$l_1^{(m)}, ..., l_d^{(m)}$]]) to represent MSG with m scales.

FC(l,dp) represents a fully connected layer with width l and dropout ratio dp. FP($l_1, ..., l_d$) is a feature propagation (FP) level with d fully connected layers. It is used for updating features concatenated from interpolation and skip link. All fully connected layers are followed by batch normalization and ReLU except for the last score prediction layer.

B.1 Network Architectures

For all classification experiments we use the following architecture (Ours SSG) with different K (number of categories):

$$SA(512, 0.2, [64, 64, 128]) \rightarrow SA(128, 0.4, [128, 128, 256]) \rightarrow SA([256, 512, 1024]) \rightarrow$$
$$FC(512, 0.5) \rightarrow FC(256, 0.5) \rightarrow FC(K)$$

The multi-scale grouping (MSG) network (PointNet++) architecture is as follows:

$$SA(512, [0.1, 0.2, 0.4], [[32, 32, 64], [64, 64, 128], [64, 96, 128]]) \rightarrow$$
$$SA(128, [0.2, 0.4, 0.8], [[64, 64, 128], [128, 128, 256], [128, 128, 256]]) \rightarrow$$
$$SA([256, 512, 1024]) \rightarrow FC(512, 0.5) \rightarrow FC(256, 0.5) \rightarrow FC(K)$$

The cross level multi-resolution grouping (MRG) network's architecture uses three branches:

Branch 1: $SA(512, 0.2, [64, 64, 128]) \rightarrow SA(64, 0.4, [128, 128, 256])$

Branch 2: $SA(512, 0.4, [64, 128, 256])$ using $r = 0.4$ regions of original points

Branch 3: $SA(64, 128, 256, 512)$ using all original points.

Branch 4: $SA(256, 512, 1024)$.

Branch 1 and branch 2 are concatenated and fed to branch 4. Output of branch 3 and branch4 are then concatenated and fed to $FC(512, 0.5) \rightarrow FC(256, 0.5) \rightarrow FC(K)$ for classification.

Network for semantic scene labeling (last two fully connected layers in FP are followed by dropout layers with drop ratio 0.5):

$$SA(1024, 0.1, [32, 32, 64]) \rightarrow SA(256, 0.2, [64, 64, 128]) \rightarrow$$
$$SA(64, 0.4, [128, 128, 256]) \rightarrow SA(16, 0.8, [256, 256, 512]) \rightarrow$$
$$FP(256, 256) \rightarrow FP(256, 256) \rightarrow FP(256, 128) \rightarrow FP(128, 128, 128, 128, K)$$

Network for semantic and part segmentation (last two fully connected layers in FP are followed by dropout layers with drop ratio 0.5):

$$SA(512, 0.2, [64, 64, 128]) \rightarrow SA(128, 0.4, [128, 128, 256]) \rightarrow SA([256, 512, 1024]) \rightarrow$$
$$FP(256, 256) \rightarrow FP(256, 128) \rightarrow FP(128, 128, 128, 128, K)$$

补充

A 概述

本补充材料提供了关于主论文中实验的更多细节，并包括更多实验来验证和分析我们提出的方法。

在 Sec B中，我们提供了用于主论文中实验的特定网络架构，并描述了数据准备和训练的细节。在 Sec C中，我们展示了更多实验结果，包括部分分割的基准性能以及邻居查询、对采样随机性的敏感性以及时空复杂性的分析。

B 实验细节

架构协议。我们使用以下符号来描述我们的网络架构。

SA(K,r,[$l_1, ..., l_d$]) 是一个集合抽象（SA）级别，使用 K 个球半径为 r 的局部区域，并使用PointNet的 d 个全连接层，宽度为 l_i ($i = 1, ..., d$). SA([$l_1, ...l_d$]) 是一个全局集合抽象级别，将集合转换为单个向量。在多尺度设置（如在MSG中），我们使用SA(K, [$r^{(1)}, ..., r^{(m)}$], [[$l_1^{(1)}, ..., l_d^{(1)}$],...,[$l_1^{(m)}, ..., l_d^{(m)}$]]) 来表示MSG，具有 m 个尺度。

FC(l,dp)表示一个宽度为 l 和dropout比率为 dp的全连接层。 FP($l_1, ..., l_d$) 是一个特征传播（FP）级别，具有 d 个全连接层。它用于更新通过插值和跳跃连接连接的特征。所有全连接层都经过批量归一化和ReLU处理，除了最后一个分数预测层。

B.1 网络架构

对于所有分类实验，我们使用以下架构（Ours SSG）并使用不同的 K(类别数量)：

$$SA(512, 0.2, [64, 64, 128]) \rightarrow SA(128, 0.4, [128, 128, 256]) \rightarrow SA([256, 512, 1024]) \rightarrow$$
$$FC(512, 0.5) \rightarrow FC(256, 0.5) \rightarrow FC(K)$$

多尺度分组（MSG）网络（PointNet++）的架构如下：

$$SA(512, [0.1, 0.2, 0.4], [[32, 32, 64], [64, 64, 128], [64, 96, 128]]) \rightarrow$$
$$SA(128, [0.2, 0.4, 0.8], [[64, 64, 128], [128, 128, 256], [128, 128, 256]]) \rightarrow$$
$$SA([256, 512, 1024]) \rightarrow FC(512, 0.5) \rightarrow FC(256, 0.5) \rightarrow FC(K)$$

跨层多分辨率分组（MRG）网络的架构使用三个分支：

分支 1： $SA(512, 0.2, [64, 64, 128]) \rightarrow SA(64, 0.4, [128, 128, 256])$

分支 2： $SA(512, 0.4, [64, 128, 256])$ 使用 $r = 0.4$ 个原始点的区域

分支 3：$SA(64, 128, 256, 512)$ 使用所有原始点。

分支 4：$SA(256, 512, 1024)$。

分支 1 和分支 2 被连接并输入到分支 4。分支 3 和分支 4 的输出然后被连接并输入到 $FC(512, 0.5) \rightarrow FC(256, 0.5) \rightarrow FC(K)$ 进行分类。

用于语义场景标注的网络（FP 中的最后两层全连接层后跟 dropout 层，dropout 比例为 0.5）：

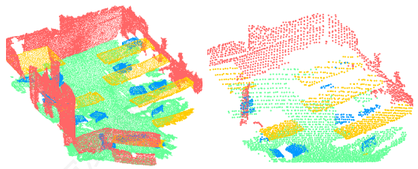
$$SA(1024, 0.1, [32, 32, 64]) \rightarrow SA(256, 0.2, [64, 64, 128]) \rightarrow$$
$$SA(64, 0.4, [128, 128, 256]) \rightarrow SA(16, 0.8, [256, 256, 512]) \rightarrow$$
$$FP(256, 256) \rightarrow FP(256, 256) \rightarrow FP(256, 128) \rightarrow FP(128, 128, 128, 128, K)$$

用于语义和部分分割的网络（FP中最后两个全连接层后跟dropout层，dropout比率为0.5）：

$$SA(512, 0.2, [64, 64, 128]) \rightarrow SA(128, 0.4, [128, 128, 256]) \rightarrow SA([256, 512, 1024]) \rightarrow$$
$$FP(256, 256) \rightarrow FP(256, 128) \rightarrow FP(128, 128, 128, 128, K)$$

B.2 Virtual Scan Generation

In this section, we describe how we generate labeled virtual scan with non-uniform sampling density from ScanNet scenes. For each scene in ScanNet, we set camera location $1.5m$ above the centroid of the floor plane and rotate the camera orientation in the horizontal plane evenly in 8 directions. In each direction, we use a image plane with size $100px$ by $75px$ and cast rays from camera through each pixel to the scene. This gives a way to select visible points in the scene. We could then generate 8 virtual scans for each test scene similar and an example is shown in Fig. 9. Notice point samples are denser in regions closer to the camera.



(a) ScanNet labeled scene (b) ScanNet non-uniform

Figure 9: Virtual scan generated from ScanNet

B.3 MNIST and ModelNet40 Experiment Details

For MNIST images, we firstly normalize all pixel intensities to range $[0, 1]$ and then select all pixels with intensities larger than 0.5 as valid digit pixels. Then we convert digit pixels in an image into a 2D point cloud with coordinates within $[-1, 1]$, where the image center is the origin point. Augmented points are created to add the point set up to a fixed cardinality (512 in our case). We jitter the initial point cloud (with random translation of Gaussian distribution $\mathcal{N}(0, 0.01)$ and clipped to 0.03) to generate the augmented points. For ModelNet40, we uniformly sample N points from CAD models surfaces based on face area.

For all experiments, we use Adam [9] optimizer with learning rate 0.001 for training. For data augmentation, we randomly scale object, perturb the object location as well as point sample locations. We also follow [21] to randomly rotate objects for ModelNet40 data augmentation. We use Tensor-Flow and GTX 1080, Titan X for training. All layers are implemented in CUDA to run GPU. It takes around 20 hours to train our model to convergence.

B.4 ScanNet Experiment Details

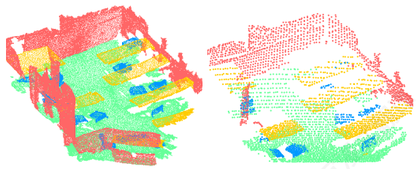
To generate training data from ScanNet scenes, we sample 1.5m by 1.5m by 3m cubes from the initial scene and then keep the cubes where $\geq 2\%$ of the voxels are occupied and $\geq 70\%$ of the surface voxels have valid annotations (this is the same set up in [5]). We sample such training cubes on the fly and random rotate it along the up-right axis. Augmented points are added to the point set to make a fixed cardinality (8192 in our case). During test time, we similarly split the test scene into smaller cubes and get label prediction for every point in the cubes first, then merge label prediction in all the cubes from a same scene. If a point get different labels from different cubes, we will just conduct a majority voting to get the final point label prediction.

B.5 SHREC15 Experiment Details

We randomly sample 1024 points on each shape both for training and testing. To generate the input intrinsic features, we to extract 100 dimensional WKS, HKS and multiscale Gaussian curvature respectively, leading to a 300 dimensional feature vector for each point. Then we conduct PCA to reduce the feature dimension to 64. We use a 8 dimensional embedding following [23] to mimic the geodesic distance, which is used to describe our non-Euclidean metric space while choosing the point neighborhood.

B.2 虚拟 ScanGeneration

在本节中，我们描述了如何从ScanNet场景中生成具有非均匀采样密度的带标签虚拟扫描。对于ScanNet中的每个场景，我们将相机位置 $1.5m$ 设置在地板平面的质心上方，并在水平平面中均匀地将相机方向旋转到8个方向。在每个方向上，我们使用一个大小为 $100px$ x $75px$ 的图像平面，并从相机通过每个像素向场景投射光线。这提供了一种选择场景中可见点的方法。然后我们可以为每个测试场景生成8个相似的虚拟扫描，一个示例显示在图9中。注意，靠近相机的区域中的点样本更密集。



(a) ScanNet 标注场景 (b) ScanNet 非均匀

图 9：从 ScanNet 生成的虚拟扫描

B.3 MNIST 和 ModelNet40 实验细节

对于 MNIST 图像，我们首先将所有像素强度归一化到范围 $[0, 1]$，然后选择所有强度大于 0.5 的像素作为有效数字像素。然后我们将图像中的数字像素转换为坐标在 $[-1, 1]$ 内的 2D 点云，其中图像中心是原点。我们创建增强点以将点集增加到固定的基数（我们的情况是 512）。我们对初始点云进行抖动（使用高斯分布 $\mathcal{N}(0, 0.01)$ 的随机平移并裁剪到 0.03）以生成增强点。对于 ModelNet40，我们根据面面积从 CAD 模型表面均匀采样 N 个点。

对于所有实验，我们使用 Adam [9] 优化器，学习率为 0.001 进行训练。对于数据增强，我们随机缩放对象，扰动对象位置以及点采样位置。我们还遵循 [21] 来随机旋转对象，用于 ModelNet40 数据增强。我们使用 Tensor-Flow 和 GTX 1080、Titan X 进行训练。所有层都使用 CUDA 实现，以运行 GPU。训练我们的模型到收敛大约需要 20 小时。

B.4 ScanNetExperiment Details

为了从 ScanNet 场景中生成训练数据，我们从初始场景中采样 1.5m x 1.5m x 3m 的立方体，然后保留其中 $\geq 2\%$ 的体素被占用且 $\geq 70\%$ 的表面体素有有效标注（这与 [5] 中的设置相同）。我们动态采样这样的训练立方体，并沿上-右轴随机旋转它。将增强点添加到点集中，以使其具有固定的基数（我们情况下的基数为 8192）。在测试时，我们同样将测试场景分割成更小的立方体，首先为每个立方体中的点获取标签预测，然后合并来自同一场景的所有立方体的标签预测。如果一个点从不同的立方体中获取到不同的标签，我们将进行多数投票以获得最终的点标签预测。

B.5 SHREC15Experiment 详细信息

我们在每个形状上随机采样1024个点，用于训练和测试。为了生成输入的内在特征，我们分别提取100维的WKS、HKS和多尺度高斯曲率，从而得到每个点的300维特征向量。然后我们进行PCA降维，将特征维度降至64。我们使用一个8维嵌入 [23]来模拟测地距离，该距离用于描述我们的非欧几里得度量空间，同时在选择点邻域时使用。

C More Experiments

In this section we provide more experiment results to validate and analyze our proposed network architecture.

C.1 Semantic Part Segmentation

Following the setting in [32], we evaluate our approach on part segmentation task assuming category label for each shape is already known. Taken shapes represented by point clouds as input, the task is to predict a part label for each point. The dataset contains 16,881 shapes from 16 classes, annotated with 50 parts in total. We use the official train test split following [4].

We equip each point with its normal direction to better depict the underlying shape. This way we could get rid of hand-crafted geometric features as is used in [32, 33]. We compare our framework with traditional learning based techniques [32], as well as state-of-the-art deep learning approaches [20, 33] in Table 4. Point intersection over union (IoU) is used as the evaluation metric, averaged across all part classes. Cross-entropy loss is minimized during training. On average, our approach achieves the best performance. In comparison with [20], our approach performs better on most of the categories, which proves the importance of hierarchical feature learning for detailed semantic understanding. Notice our approach could be viewed as implicitly building proximity graphs at different scales and operating on these graphs, thus is related to graph CNN approaches such as [33]. Thanks to the flexibility of our multi-scale neighborhood selection as well as the power of set operation units, we could achieve better performance compared with [33]. Notice our set operation unit is much simpler compared with graph convolution kernels, and we do not need to conduct expensive eigen decomposition as opposed to [33]. These make our approach more suitable for large scale point cloud analysis.

	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
Yi [32]	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
PN [20]	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
SSCNN [33]	84.7	81.6	81.7	81.9	75.2	90.2	74.9	93.0	86.1	84.7	95.6	66.7	92.7	81.6	60.6	82.9	82.1
Ours	85.1	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

Table 4: Segmentation results on ShapeNet part dataset.

C.2 Neighborhood Query: kNN v.s. Ball Query.

Here we compare two options to select a local neighborhood. We used radius based ball query in our main paper. Here we also experiment with kNN based neighborhood search and also play with different search radius and k. In this experiment all training and testing are on ModelNet40 shapes with uniform sampling density. 1024 points are used. As seen in Table 5, radius based ball query is slightly better than kNN based method. However, we speculate in very non-uniform point set, kNN based query will results in worse generalization ability. Also we observe that a slightly large radius is helpful for performance probably because it captures richer local patterns.

kNN (k=16)	kNN (k=64)	radius (r=0.1)	radius (r=0.2)
89.3	90.3	89.1	90.7

Table 5: Effects of neighborhood choices. Evaluation metric is classification accuracy (%) on ModelNet 40 test set.

C.3 Effect of Randomness in Farthest Point Sampling.

For the *Sampling layer* in our set abstraction level, we use farthest point sampling (FPS) for point set sub sampling. However FPS algorithm is random and the subsampling depends on which point is selected first. Here we evaluate the sensitivity of our model to this randomness. In Table 6, we test our model trained on ModelNet40 for feature stability and classification stability.

更多实验

在本节中，我们提供更多实验结果来验证和分析我们提出的网络架构。

C.1 语义部分分割

根据 [32], 中的设置，我们在假设每个形状的类别标签已知的部分分割任务上评估了我们的方法。以点云表示的形状作为输入，任务是预测每个点的部分标签。该数据集包含来自 16 个类别的 16,881 个形状，总共标注了 50 个部分。我们遵循 [4] 的官方训练测试划分。

我们为每个点配备了其法线方向，以更好地描绘底层形状。这样我们就可以摆脱在 [32, 33] 中使用的手工几何特征。我们在表 4 中将我们的框架与传统的基于学习的 [32], 技术以及最先进的深度学习方法 [20, 33] 进行了比较。点交集重叠 (IoU) 被用作评估指标，在所有部分类别上取平均值。在训练过程中最小化交叉熵损失。平均而言，我们的方法取得了最佳性能。与 [20], 相比，我们的方法在大多数类别上都表现更好，这证明了层次特征学习对于详细的语义理解的重要性。请注意，我们的方法可以被视为在多个尺度上隐式构建邻近图并在这些图上操作，因此与 [33] 等图 CNN 方法相关。得益于我们多尺度邻域选择的可扩展性以及集合运算单元的强大功能，我们能够与 [33] 相比取得更好的性能。请注意，我们的集合运算单元与图卷积核相比要简单得多，并且我们不需要进行昂贵的特征分解，这与 [33] 不同。这些使得我们的方法更适合大规模点云分析。

	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	手枪	rocke t	滑板	桌子
Yi [32]	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
PN [20]	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
SSCNN [33]	84.7	81.6	81.7	81.9	75.2	90.2	74.9	93.0	86.1	84.7	95.6	66.7	92.7	81.6	60.6	82.9	82.1
Ours	85.1	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

表4：ShapeNet部件数据集上的分割结果。

C.2 NeighborhoodQuery: kNNv.s. Ball Query。

在这里，我们比较了两种选择局部邻域的方案。我们在主要论文中使用了基于半径的球查询。在这里，我们也对基于kNN的邻域搜索进行了实验，并尝试了不同的搜索半径和 k。在这个实验中，所有训练和测试都在具有均匀采样密度的ModelNet40形状上进行。使用了1024个点。如表5所示，基于半径的球查询略好于基于kNN的方法。然而，我们推测在非常非均匀的点集中，基于kNN的查询会导致泛化能力更差。此外，我们观察到略微较大的半径对性能有帮助，可能是因为它捕获了更丰富的局部模式。

kNN (k=16)	kNN (k=64)	半径 (r=0.1)	半径 (r=0.2)
89.3	90.3	89.1	90.7

表 5：邻域选择的影响。评估指标是在 ModelNet 40 测试集上的分类准确率 (%)。

C.3 最远点采样中的随机性影响。

对于我们的集合抽象层中的 采样层 ，我们使用最远点采样（FPS）进行点集子采样。然而 FPS 算法是随机的，子采样取决于首先选择哪个点。在这里，我们评估我们的模型对此随机性的敏感性。在表 6 中，我们测试我们在 ModelNet40 上训练的模型在特征稳定性和分类稳定性方面的表现。

To evaluate feature stability we extract global features of all test samples for 10 times with different random seed. Then we compute mean features for each shape across the 10 sampling. Then we compute standard deviation of the norms of feature's difference from the mean feature. At last we average all std. in all feature dimensions as reported in the table. Since features are normalized into 0 to 1 before processing, the 0.021 difference means a 2.1% deviation of feature norm.

For classification, we observe only a 0.17% standard deviation in test accuracy on all ModelNet40 test shapes, which is robust to sampling randomness.

Feature difference std.	Accuracy std.
0.021	0.0017

Table 6: Effects of randomness in FPS (using ModelNet40).

C.4 Time and Space Complexity.

Table 7 summarizes comparisons of time and space cost between a few point set based deep learning method. We record forward time with a batch size 8 using TensorFlow 1.1 with a single GTX 1080. The first batch is neglected since there is some preparation for GPU. While PointNet (vanilla) [20] has the best time efficiency, our model without density adaptive layers achieved smallest model size with fair speed.

It's worth noting that ours MSG, while it has good performance in non-uniformly sampled data, it's 2x expensive than SSG version due the multi-scale region feature extraction. Compared with MSG, MRG is more efficient since it uses regions across layers.

	PointNet (vanilla) 1	PointNet 1	Ours (SSG)	Ours (MSG)	Ours (MRG)
Model size (MB)	9.4	40	8.7	12	24
Forward time (ms)	11.6	25.3	82.4	163.2	87.0

Table 7: Model size and inference time (forward pass) of several networks.

为了评估特征稳定性，我们使用不同的随机种子对所有测试样本提取全局特征 10 次。然后，我们对每个形状在 10 次采样中计算平均特征。接着，我们计算特征与平均特征之差的范数的标准差。最后，我们将所有特征维度中的所有 std. 平均，如表中所示。由于特征在处理前被归一化到 0 到 1 之间，0.021 的差异意味着特征范数有 2.1% 的偏差。

对于分类，我们在所有 ModelNet40 测试形状上观察到的测试准确率标准差仅为 0.17%，且对采样随机性具有鲁棒性。

特征差异标准差	准确率标准差
0.021	0.0017

表 6：FPS 中随机性的影响（使用 ModelNet40）。

C.4 时间和空间复杂度。

表7总结了几个基于点集的深度学习方法在时间和空间成本上的比较。我们使用TensorFlow 1.1和单个GTX 1080记录了批量大小为8的前向时间。由于GPU需要一些准备，所以第一个批次被忽略。虽然PointNet（原版） [20]具有最佳的时间效率，但我们的模型在不使用密度自适应层的情况下实现了最小的模型尺寸和合理的速度。

值得注意的是，我们的MSG在非均匀采样数据中表现良好，但由于多尺度区域特征提取，它的成本比SSG版本高2倍。与MSG相比，MRG更高效，因为它使用跨层的区域。

	PointNet（原版）1	PointNet 1	ours (SSG)	Ours (MSG)	Ours (MRG)
模型大小 (MB)	9.4	40	8.7	12	24
前向时间 (ms)	11.6	25.3	82.4	163.2	87.0

表 7：几个网络模型的模型大小和推理时间（前向传递）。