

# PointNet++: メトリック空間上の点集合に対する深い階層的特徴学習

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

## Abstract

点集合に対する深層学習を研究した先行研究はほとんどない。PointNet [20]はこの方向でのパイオニアである。しかし、PointNetは設計上、点が住むメトリック空間によって誘発される局所構造を捕捉しないため、細かいパターンを認識する能力と複雑なシーンへの汎化性が制限されている。そこで本研究では、入力点群に対して再帰的にPointNetを適用する、階層型ニューラルネットワークを導入する。計量空間距離を利用することで、我々のネットワークは文脈のスケールが大きくなるにつれて、局所的な特徴を学習することができる。さらに、点集合は通常様々な密度でサンプリングされるため、均一な密度で学習したネットワークの性能は大きく低下する。そこで我々は、複数のスケールからの特徴を適応的に結合する新しい集合学習層を提案する。実験により、我々が提案するPointNet++と呼ばれるネットワークは、深い点集合の特徴を効率的かつ頑健に学習することができることが示された。特に、3次元点群の困難なベンチマークにおいて、最先端技術を大幅に上回る結果が得られている。

## 1 Introduction

我々は、ユークリッド空間における点の集まりである幾何学的点集合の解析に興味がある。幾何学的点群として特に重要なのは、例えば、適切に装備された自律走行車から撮影された3Dスキャナによる点群である。このようなデータは集合として、そのメンバーの並べ替えに対して不变でなければならない。さらに、距離メトリックは、異なる特性を示す可能性のある局所的な近傍を定義する。例えば、点の密度やその他の属性は、異なる場所で一様でない場合がある。3Dスキャンでは、密度の変動は、遠近効果、放射状の密度の変動、動きなどに由来する。

点群に対する深層学習を研究している先行研究はほとんどない。PointNet[20]は点集合を直接処理する先駆的な取り組みである。PointNetの基本的な考え方は、各点の空間的なエンコーディングを学習し、次に、すべての個々の点の特徴をグローバルな点群シグネチャに集約することである。PointNetは、その設計上、メトリックによって引き起こされる局所的な構造を捕捉しない。しかし、局所的な構造を利用することは、畳み込みアーキテクチャの成功にとって重要であることが証明されている。CNNは規則正しいグリッドで定義されたデータを入力とし、多重解像度階層に沿って次第に大きなスケールで特徴を捉えることができる。低レベルのニューロンは小さな受容野を持つが、高レベルのニューロンは大きな受容野を持つ。また、階層に沿った局所的なパターンを抽出することができるため、未知のケースに対する汎化性が高い。

本論文では、計量空間でサンプリングされた点の集合を階層的に処理する、PointNet++ と名付けられた階層型ニューラルネットワークを紹介する。PointNet++ の一般的な考え方は単純である。まず、点集合を基礎となる空間の距離メトリックによって、重なり合う局所領域に分割する。CNNと同様に、局所領域から微細な幾何学的構造を捉えた局所特徴を抽出し、その局所特徴をさらに大きな単位にグループ化し、高次の特徴を生成する処理を行う。この処理を繰り返すことで、点群全体の特微量を得ることができる。

PointNet++の設計では、点集合の分割をどのように行うか、および、局所特徴学習器を用いて点集合や局所特徴をどのように抽象化するかという二つの課題に取り組む必要がある。この2つの課題

---

# 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

といふのは、点集合の分割は、畳み込みの設定のように、局所特徴学習器の重みを共有できるように、分割間で共通の構造を生成しなければならないからである。我々は局所特徴学習器としてPointNetを選択した。PointNetは、意味的特徴抽出のための非順序点集合の処理に有効なアーキテクチャであることが、この研究で示された。さらに、このアーキテクチャは入力データの破損に対してロバストである。PointNetは、基本的な構成要素として、局所的な点の集合や特徴をより高次の表現に抽象化する。PointNet++は、入力集合のネストされたパーティショニングに対して再帰的にPointNetを適用する。

PointNet++では、点群の重複する分割をどのように生成するかという問題が残されています。各分割はユークリッド空間における近傍球として定義され、そのパラメータにはセントロイドの位置とスケールが含まれる。点集合全体を均等にカバーするために、遠点サンプリング(FPS)アルゴリズムによって入力点集合からセントロイドが選択される。図1: 走査距離固定で空間を走査するvolumetric CNNと比較して、我々の局所受容野はStructure Sensor(左:RGB、右:点群)に依存しており、入力データとメトリックの両方に依存するため、より効率的かつ効果的である。

しかし、局所近傍球の適切なスケールを決定することは、特徴のスケールと入力点集合の非一様性のもつれにより、より困難かつ興味深い問題である。我々は、入力点群が異なる領域で可変の密度を持つと仮定する。これは、構造センサースキャン[18]のような実データで非常に一般的である(図1参照)。このように、我々の入力点セットは、一定の密度を持つ規則的なグリッド上に定義されたデータとみなすことができるCNNの入力とは大きく異なるものである。CNNにおいて、局所的なパーティション・スケールに対応するのはカーネルのサイズである。[25]はより小さなカーネルを使うことがCNNの能力を向上させることを示している。しかしながら、点集合データに対する我々の実験はこの法則に反する証拠を与える。小さな近傍領域は、サンプリング不足のためにポイントが少なすぎ、ポイントネットがパターンをロバストに捕らえるには不十分である可能性がある。

PointNet++は、複数のスケールの近傍領域を利用して、ロバスト性と細部の捕捉の両方を実現したことが、本論文の重要な貢献です。PointNet++は、学習中のランダムな入力ドロップアウトを利用して、異なるスケールで検出されたパターンを適応的に重み付けし、入力データに応じてマルチスケール特徴を結合することを学習する。実験によると、我々のPointNet++は点群データを効率的かつ頑健に処理することができる事が示された。特に、3次元点群の困難なベンチマークに対して、最先端技術を大きく上回る結果が得られている。

## 2 Problem Statement

$X = (M, d)$  はユークリッド空間  $R^n$  のメトリックを継承した離散メトリック空間で、 $M \subseteq R^n$  は点集合、 $d$  は距離メトリックであるとする。また、アンビエントユークリッド空間における  $M$  の密度はどこでも一様でない場合がある。我々は、このような  $X$  を入力として(各点の追加的な特徴とともに)、 $X$  を評価して意味的に興味深い情報を生成する集合関数  $f$  を学習することに専念する。実際には、このような  $f$  は、 $X$  にラベルを割り当てる分類関数や、 $M$  の各メンバーに点ごとのラベルを割り当てるセグメンテーション関数とすることができます。

## 3 Method

本論文では、PointNet [20]に階層構造を追加した拡張版と見なすことができる。まず PointNet について概説し(第 3.1 節)、次に PointNet に階層構造を付加した基本的な拡張を紹介する(第 3.2 節)。最後に、サンプリングが一様でない点群に対しても頑健に特徴量を学習できる我々の PointNet++ を提案する(Sec. 3.3)。

### 3.1 PointNet のレビュー[20]。汎用連続集合関数近似器

$x_1, x_2, \dots, x_n$  の順不同の点集合  $\{x_1, x_2, \dots, x_n\}$  が与えられると、点集合をベクトルに写像する集合関数  $f : X \rightarrow R$  を定義することができる。

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

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.

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

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



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

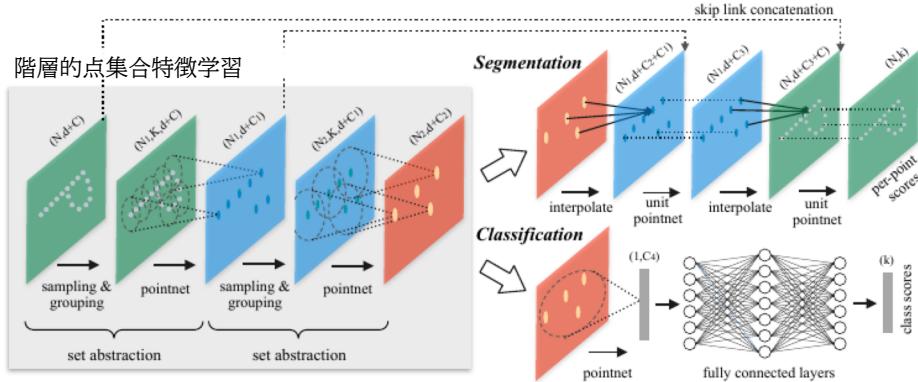


図 2: 階層的特徴学習アーキテクチャとその応用例として、2次元ユークリッド空間の点を用いた集合分割・分類の説明図である。ここでは、シングルスケール点のグルーピングを可視化した。密度適応的なグルーピングについては、図3を参照。

ここで、 $\gamma$  と  $h$  は通常 MLP (Multi-layer perceptron) ネットワークである。

式1の集合関数  $f$  は入力点の並べ替えに対して不变であり、任意の連続集合関数を任意に近似することができる[20]。なお、 $h$  の応答は点の空間エンコード38として解釈できる（詳細は[20]を参照）。

PointNet はいくつかのベンチマークで印象的な性能を達成しました。しかし、異なるスケールでの局所的な文脈を捉える能力に欠ける。この制限を解決するために、次節で階層的な特徴学習フレームワークを導入する予定である。

### 3.2 階層的な点群特徴量学習

PointNet では、1回の最大プーリング操作で全点集合を集約するが、我々の新しいアーキテクチャでは、点の階層的なグループ化を行い、階層に沿ってより大きな局所領域を徐々に抽出していく。

我々の階層的構造は、いくつかの集合抽象化レベルによって構成される（図2）。各階層では、点の集合を処理し、より少ない要素で新しい集合を生成する抽象化を行う。集合抽象化レベルは、3つの主要な層で構成されている。サンプリング層、グルーピング層、ポイントネット層である。サンプリング層は、入力点から点の集合を選択し、局所領域の中心点を定義する。グルーピング層は、入力点から局所領域の中心点を抽出し、その中心点周辺の「近傍点」を見つけることで局所領域群を構成する。PointNet 層は、ミニ PointNet を用いて、局所領域のパターンを特徴ベクトルにエンコードする。

集合抽象化レベルは、 $d$ -dim 座標と  $C$ -dim 点特徴を持つ  $N$  個の点からなる  $N \times (d+C)$  行列を入力として受け取る。これは、 $d$ -dim 座標を持つ  $N$  個のサブサンプル点の  $N \times (d+C)$  行列と、局所コンテキストを要約した新しい  $C$ -dim 特徴ベクトルを出力する。以下に、集合の抽象化レベルの階層を紹介する。

**サンプリング層。** 入力点  $\{x_1, x_2, \dots, x_n\}$  が与えられたとき、反復的遠点サンプリング (FPS) を用いて、残りの点に関して  $x_i$  が集合  $\{x_{i1}, x_{i2}, \dots, x_{im}\}$  から最も遠い点（メートル距離で）であるような点の部分集合  $\{x_{i1}, x_{i2}, \dots, x_{i,j-1}\}$  を選択する。ランダムサンプリングと比較すると、同じ数のセントロイドがあれば、点集合全体をよりよくカバーすることができる。CNNがデータ分布によらずベクトル空間を走査するのとは対照的に、我々のサンプリング戦略はデータに依存した方法で受容野を生成する。

**グループ化層。** この層への入力はサイズ  $N \times (d+C)$  の点集合とサイズ  $N \times d$  のセントロイドの集合の座標である。出力はサイズ  $N \times K \times (d+C)$  の点集合のグループであり、各グループは局所領域に対応し  $K$  はセントロイド点の近隣にある点の個数である。 $K$  はグループによって異なるが、後続の PointNet 層は柔軟な点数を固定長の局所領域特徴ベクトルに変換することができることに注意。

畳み込みニューラルネットワークでは、ある画素の局所領域は、その画素からあるマンハッタン距離（カーネルサイズ）以内にある配列インデックスを持つ画素から構成される。メトリックス空間からサンプリングされた点集合において、点の近傍領域はメトリックス距離で定義される。

ボールクエリは、クエリ点から半径の範囲内にあるすべての点を見つける（実装では  $K$  の上限が設定される）。別の範囲検索として、 $K$  最近傍 ( $kNN$ ) 検索があり、この検索では固定された

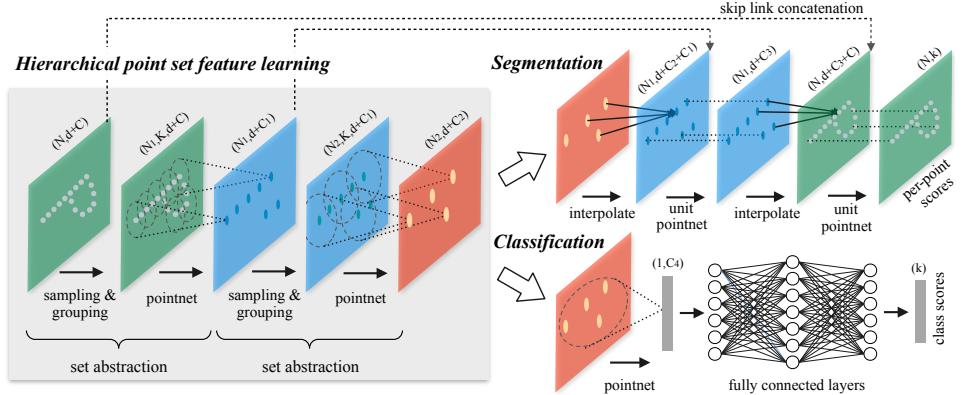


Figure 2: Illustration of our hierarchical feature learning architecture and its application for set segmentation and classification using points in 2D Euclidean space as an example. Single scale point grouping is visualized here. For details on density adaptive grouping, see Fig. 3

where  $\gamma$  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  $\{x_1, x_2, \dots, x_n\}$ , we use iterative farthest point sampling (FPS) to choose a subset of points  $\{x_{i_1}, x_{i_2}, \dots, x_{i_m}\}$ , such that  $x_{i_j}$  is the most distant point (in metric distance) from the set  $\{x_{i_1}, x_{i_2}, \dots, x_{i_{j-1}}\}$  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

近傍点の数は一定である。kNNと比較して、ボールクエリの局所近傍は一定の領域スケールを保証するため、局所領域特徴は空間全体でより一般化され、局所パターン認識を必要とするタスク（例：セマンティックポイントラベリング）に好適である。

PointNet 層。この層では、データサイズ  $N \times K \times (d+C)$  の点からなる  $N$  個の局所領域が入力とされる。出力における各局所領域は、そのセントロイドと、セントロイドの近傍を符号化した局所特徴量によって抽象化される。出力データサイズは  $N \times (d + C)$  である。

局所領域内の点の座標は、まずセントロイドの点を基準にロー カルフレームに変換される： $x_i = x_i - x^*(j)$  for  $i = 1, 2, \dots, K$  and  $j = 1, 2, \dots, d$  where  $x^*$  はセントロイドの座標 ( $j$ ) ( $j$ ) である。局所パターン学習の基本構成要素として、3.1 節で述べた PointNet[20]を使用する。点特徴量とともに相対座標を用いることで、局所領域における点と点の関係を捉えることができる。

### 3.3 不均一なサンプリング密度の下での頑健な特徴学習

前述したように、点集合は地域によって密度が不均一であることが一般的である。このような非一様性は、点集合の特徴学習にとって重要な課題をもたらす。密なデータで学習した特徴は、疎なサンプルの領域には汎化できないかもしれない。その結果、疎な点群に対して学習したモデルは、細かな局所構造を認識できない可能性がある。

理想的には、点群に対して可能な限り詳細に検査したい  
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 そこで我々は、入力サンプリング密度が変化した際に、異なるスケールの領域からの特徴を組み合わせることを学習する、密度適応型ポイントネット層（図3）を提案する。この密度適応型 PointNet 層を用いた階層型ネットワークを PointNet++ と呼ぶことにする。

前節（3.2）では、各抽出レベルには一つのスケールのグループ化と特徴抽出が含まれていた。PointNet++ では、各抽出レベルにおいて、複数のスケールの局所パターンを抽出し、局所点密度に応じて、それらをインテリジェントに結合する。局所領域のグループ化と異なるスケールの特徴の組み合わせに関して、我々は以下のような2種類の密度適応層を提案する。

マルチスケールグループ化 (MSG)。図 3(a)に示すように、マルチスケールパターンを捉えるシンプルで効果的な方法は、異なるスケールを持つグループ化レイヤーを適用し、その後、各スケールの特徴を抽出するポイントネットに従うことである。異なるスケールの特徴を連結し、マルチスケール特徴を形成する。

マルチスケール特徴を結合するための最適な戦略を学習するために、ネットワークを学習させる。これは、各インスタンスに対してランダムな確率で入力点をドロップアウトさせることにより行われる（これをランダムインバットドロップアウトと呼ぶ）。具体的には、各トレーニング点セットに対して、 $[0, p]$  から一様にサンプリングしたドロップアウト率  $\theta$  を選択する ( $p \leq 1$ )。各点に対して、確率  $\theta$  でランダムに点を落とす。実際には、空の点セットの生成を避けるために、 $p = 0.95$  に設定する。そうすることで、（ $\theta$  によって引き起こされる）様々なスペース性と（ドロップアウトのランダム性によって引き起こされる）様々な均一性のトレーニングセットをネットワークに提示することができる。テストの間、我々はすべての利用可能なポイントを保持する。

多重解像度グループ化 (MRG)。上記のMSGアプローチは、すべてのセントロイド点に対して大規模な近傍でローカルPointNetを実行するため、計算量が多くなります。特に、最下層ではセントロイド点の数がかなり多くなるのが普通であるため、時間的なコストが大きくなる。

ここでは、このような高価な計算を回避しつつ、点の分布特性に応じて適応的に情報を集約する機能を保持した代替手法を提案する。図 3(b)において、あるレベル  $L_i$  における領域の特徴は、2 つのベクトルの連結である。1 つのベクトル（図中左）は、下位レベル  $L_{i-1}$  から各小領域の特徴を集合抽象度によりまとめたものである。もう一方のベクトル（図中右）は、局所領域内の全ての生点を 1 つの PointNet で直接処理することで得られる特徴量である。

局所領域の密度が低い場合、1 番目のベクトルを計算する部分領域は、さらに疎な点を含み、サンプリング不足に陥るため、2 番目のベクトルよりも信頼性が低くなることがある。このような場合は、第二のベクトルをより高く重み付けする必要がある。一方

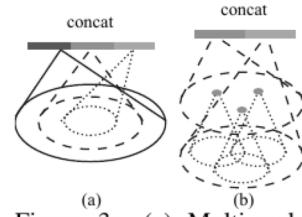


Figure 3: (a) Multi-scale grouping (MSG); (b) Multi-resolution grouping (MRG).

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  $i = 1, 2, \dots, K$  and  $j = 1, 2, \dots, d$  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.

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  $\theta$  uniformly sampled from  $[0, p]$  where  $p \leq 1$ . For each point, we randomly drop a point with probability  $\theta$ . 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  $\theta$ ) 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,

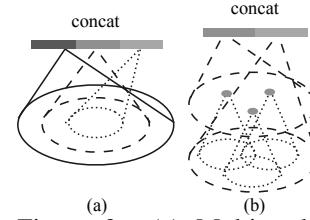


Figure 3: (a) Multi-scale grouping (MSG); (b) Multi-resolution grouping (MRG).

局所領域の密度が高い場合、第一のベクトルは低レベルで再帰的に高解像度で検査する能力を持つため、より詳細な情報を提供することができます。

本手法は、MSGと比較して、大規模な近傍領域の特徴抽出を最下層で回避できるため、計算効率が良い。

### 3.4 集合分割のための点特微量伝搬法

集合抽象化層では、元の点集合をサブサンプリングする。しかし、意味的な点ラベリングのような集合分割のタスクでは、すべての元の点に対する点特徴を得たい。一つの解決策は、全ての集合抽象化レベルにおいて、常に全ての点をセントロイドとしてサンプリングすることであるが、これは高い計算コストとなる。また、サブサンプリングされた点から元の点へ特徴を伝搬させる方法もある。

図2に示すように、距離ベースの補間とレベル間のスキップリンクを用いた階層的な伝搬戦略を採用する。ここで、 $N_{l-1}$ と $N_l$  ( $N_l \subset N_{l-1}$ ) は集合抽象化レベル  $l$  の入力と出力の点集合サイズである。補間方法には多くの選択肢があるが、ここでは  $k$  個の最近傍点を用いた逆距離加重平均を用いる（式2のように、デフォルトでは  $p = 2$ ,  $k = 3$  を用いる）。次に、 $N_{l-1}$  点の補間された特微量は、集合抽象度から のスキップリンク点特微量と連結される。次に、連結された特徴は、CNNにおける一対一の畠み込みに類似した「ユニットポイントネット」に渡される。いくつかの共有完全接続層とReLU層が、各点の特徴ベクトルを更新するために適用される。このプロセスは、元の点群に特徴を伝搬させるまで繰り返される。

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

## 4 Experiments

データセット 我々は、2Dオブジェクト (MNIST [11])、3Dオブジェクト (ModelNet40 [31] rigid object, SHREC15 [12] non-rigid object)、実際の3Dシーン (ScanNet [5]) までの4つのデータセットで評価します。オブジェクトの分類は精度で評価する。また、シーンラベル付けは、[5]に従い、ボクセルの平均分類精度で評価する。以下に、各データセットの実験設定を示す。

- MNIST: MNIST: 手書き数字画像、60kトレーニングサンプルと10kテストサンプル。
- ModelNet40: 40種類のCADモデル（主に人工物）。トレーニング用形状9,843個、テスト用形状2,468個と公式の分割を使用。
- SHREC15: 50カテゴリから1200個の形状。各カテゴリには24個の形状が含まれ、それらは馬や猫など様々なポーズを持つ有機的なものがほとんどである。このデータセットでは、5重クロスバリデーションを用いて、分類精度を獲得している。
- ScanNet: 1513点の室内シーンをスキャンし、再構築したデータセット。このデータセットでは、[5]の実験設定に従い、トレーニングに1201シーン、テストに312シーンを使用する。

### 4.1 ユークリッドメトリック空間における点群分類

2次元 (MNIST) と3次元 (ModelNet40) のユークリッド空間からサンプリングした点群の分類について、我々のネットワークを評価した。MNISTの画像は、2次元の点群に変換され、各ピクセルの位置が示されている。3次元点群は、ModelNet40の形状からメッシュ面をサンプリングしたものである。デフォルトでは、MNISTは512点、ModelNet40は1024点である。表2の最後の行 (ours normal) では、顔の法線を追加点特徴として使用しており、さらに性能を上げるためにより多くの点 ( $N = 5000$ ) を使用している。すべての点集合は、ゼロ平均で単位球内になるように正規化されている。我々は、3つの完全接続された層を持つ3レベルの階層型ネットワークを使用する1結果。表1、表2において、我々の手法と過去の代表的な最先端技術のセットを比較した。なお、表2の PointNet (vanilla) は [20] の変換ネットワークを用いないバージョンであり、我々の1階層のみの階層型ネットと同等である。

まず、我々の階層型学習アーキテクチャは、非階層型PointNet[20]よりも大幅に優れた性能を達成していることがわかる。MNISTでは、相対的に 60.8%、34.6% のエラー率削減が確認されている。

ネットワークアーキテクチャや実験準備の詳細については、補足を参照。

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.

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

## 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 (ModelNet40) 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<sup>1</sup>

**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.

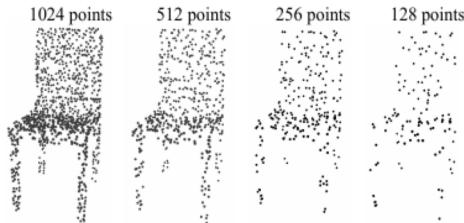
Firstly, our hierarchical learning architecture achieves significantly better performance than the non-hierarchical PointNet [20]. In MNIST, we see a relative 60.8% and 34.6% error rate reduction

---

<sup>1</sup>See supplementary for more details on network architecture and experiment preparation.

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

表 1: MNIST の数字分類



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	<b>91.9</b>

表 2: ModelNet40 形状分類

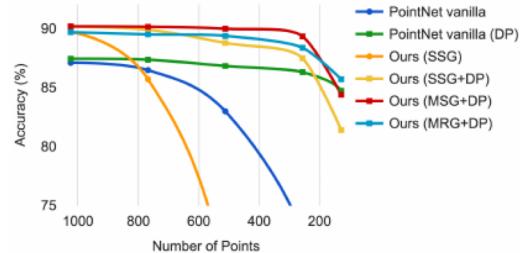


図4: 左: ランダムな点脱落を伴う点群。右図 図4 左: ランダムな点の脱落がある点群 右: 密度適応戦略の非一様な密度に対する優位性を示す曲線。DPは学習中のランダムな入力の脱落を意味し、それ以外は一様な密度の点で学習する。詳細は3.3節を参照。

PointNet (vanilla)と PointNet から我々の手法に変更した。また、ModelNet40の分類では、同じ入力データサイズ（1024点）と特徴量（座標のみ）を用いた場合、我々の方法がPointNetよりも顕著に強いことがわかる。次に、点群に基づく手法は、成熟した画像CNNと同等以上の性能を達成することが可能であることがわかる。MNISTでは、我々の手法（2次元点群に基づく）は、Network CNNのNetworkに近い精度を達成している。ModelNet40では、通常の情報を用いた我々の手法は、以前の最先端手法であるMVCNN[26]を大幅に凌駕する性能を示している。サンプリング密度のばらつきに対する頑健性。実世界から直接取得されたセンサデータは、通常、深刻な不規則サンプリングの問題に悩まされている（図1）。本手法では、複数のスケールの点近傍を選択し、それらを適切に重み付けすることで、記述性と頑健性のバランスをとるように学習する。

図4左のように、テスト時にランダムに点を削除し、非一様で疎なデータに対する本ネットワークの頑健性を検証した。図4右では、MSG+DP（訓練中にランダムに入力をドロップアウトするマルチスケールグレーピング）とMRG+DP（訓練中にランダムに入力をドロップアウトするマルチ解像度グレーピング）がサンプリング密度の変動に対して非常にロバストであることが分かる。MSG+DPの性能は1024テストポイントから256テストポイントまで1%未満しか低下しない。さらに、MSG+DPはほぼすべてのサンプリング密度において、代替案と比較して最も高い性能を達成します。PointNet vanilla [20]は、細かい部分よりも大域的な抽象化に重点を置いていたため、密度のばらつきに対してかなり頑健です。しかし、詳細が失われることで、我々のアプローチと比較して性能は低下します。また、SSG (ablated PointNet++ with single scale grouping in each level) は疎なサンプリング密度に対して一般化できないが、SSG+DP は学習時にランダムにポイントをドロップアウトすることで問題を改善する。

#### 4.2 セマンティックシーンラベリングのための点群セグメンテーション

本アプローチが大規模な画像処理に適していることを検証するため、画像処理に適した点群セグメンテーションを行った。

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 3DCNN[3] and PointNet[19].

に適していることを検証するために、ボクセル単位で公平な比較を行うために、全ての実験において RGB 情報を除去し、[5] に従って点群ラベル予測をボクセルラベリングに変換している。また、[20]との比較も行っている。図5では、ボクセル単位での精度を報告している（青棒）。

我々のアプローチは、全てのベースライン手法を大きなマージンをもって凌駕している。また、ボクセル化されたスキヤン画像から学習する[5]と比較して、我々は点群から直接学習することで量子化誤差を回避している。

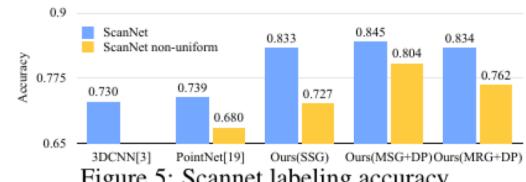
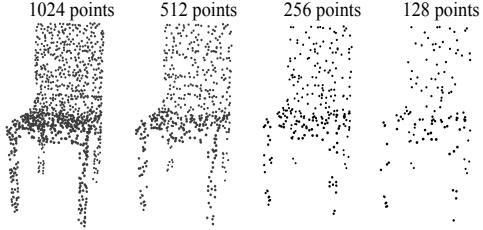


Figure 5: Scannet labeling accuracy.

Method	Error rate (%)
Multi-layer perceptron [24]	1.60
LeNet5 [11]	0.80
Network in Network [13]	<b>0.47</b>
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	<b>91.9</b>

Table 2: ModelNet40 shape classification.

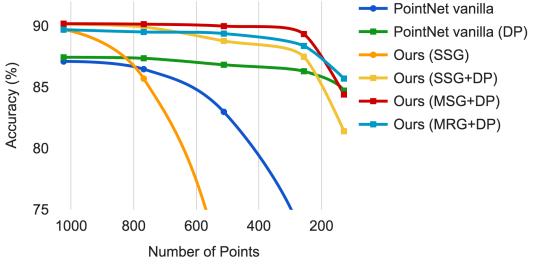


Figure 4: Left: Point cloud with random point dropout. Right: Curve showing advantage of our density adaptive strategy in dealing with non-uniform density. DP means random input dropout during training; otherwise training is on uniformly dense points. See Sec.3.3 for details.

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).

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,

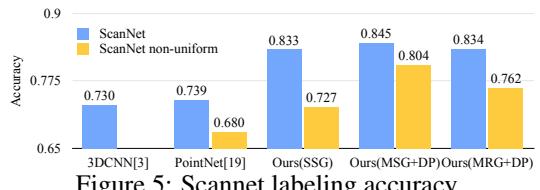


Figure 5: Scannet labeling accuracy.

また、データ依存のサンプリングを行うことで、より効果的な学習を可能としている。[20]と比較すると、我々のアプローチは階層的な特徴学習を導入し、異なるスケールでの形状特徴を捉えることができる。これは、シーンを複数のレベルで理解し、様々な大きさのオブジェクトをラベリングするために非常に重要である。図6に、シーンラベリング結果の例を示す。

サンプリング密度変動に対する頑健性 サンプリング密度が一様でないスキャンに対して、我々の学習したモデルがどのように動作するかをテストするために、図1のようなScannetシーンの仮想スキャンを合成し、このデータで我々のネットワークを評価した。仮想スキャンの生成方法については補足資料を参照されたい。また、本フレームワークを3つの設定 (SSG, MSG+DP, MRG+DP) で評価し、ベースライン手法[20]と比較した。

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 のサンプリングは疎である。本論文では、ベースライン手法[20]を比較し、密度適応型ネットワークが有効であることを示す。これらは、我々の密度適応型レイヤー設計の有効性を証明するものである。

#### 4.3 非ユークリッドメトリック空間における点群分類

本章では、本手法の非ユークリッド空間への一般化について述べる。非剛体形状分類（図7）において、優れた分類器は、図7の(a)と(c)の姿勢が異なっていても、正しく同じカテゴリとして分類できる必要があり、それには固有構造の知識が必要である。SHREC15における图形は、3次元空間に埋め込まれた2次元曲面である。表面に沿った測地線距離は、当然ながらメトリック空間を引き起こす。この計量空間において、PointNet++を採用することで、点集合の本質的な構造を捉えることができることを実験により示す。

12]の各形状について、まず、対の測地線距離によって誘起されるメトリック空間を構築する。我々は[23]に従い、測地線距離を模倣した埋め込みメトリックを得る。次に、WKS [1]、HKS [27]、マルチスケールガウス曲率 [16]など、このメトリック空間における固有の点特徴を抽出する。これらの特徴を入力として用い、基礎となる計量空間に従って点のサンプリングとグループ化を行う。このようにして、私たちのネットワークは、[1][2]の影響を受けないマルチスケールの固有構造を捉えることを学習する。図7：形状の非特異的ポーズの例。図7：形状の非特定ポーズの例。XY Z 座標を点特徴として用いるか、ユークリッド空間 R 3 を基礎的なメトリック空間として用いるかである。以下に、これらが最適な選択でないことを示す。

結果 本手法と従来の最先端手法[14]を表 3 で比較する。[14]は測地線モーメントを形状特徴量として抽出し、stacked sparse autoencoderを用いてこれらの特徴量を消化し、形状カテゴリを予測するものである。非ユークリッドメトリック空間と固有特徴を用いた我々のアプローチは、全ての設定において最高の性能を達成し、[14]を大きく上回る性能を示した。

本アプローチの第一設定と第二設定を比較すると、固有特徴が非剛体形状の分類に非常に重要であることがわかる。XY Z 特徴は、本質的な構造を明らかにできず、ポーズの変化に大きく影響される。2つ目の設定と3つ目の設定を比較すると、測地線近傍を用いることがユークリッド近傍と比較して有益であることが分かる。ユークリッド近傍は、表面上の遠方の点を含む可能性があり、形状が非剛体的な変形をする場合、この近傍は劇的に変化する可能性がある。そのため、局所構造が組合せ的に複雑になり、効果的な重み付けを行うことが難しくなる。一方、曲面上の測地線近傍は、この問題を解決し、学習効果を向上させる。

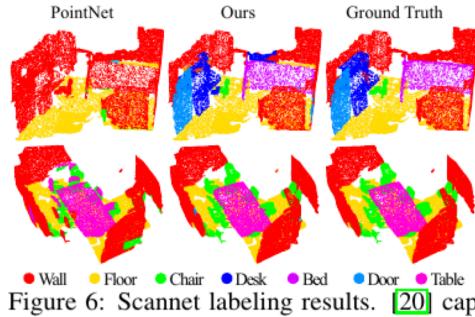


Figure 6: Scannet labeling results. [20] captures the overall layout of the room correctly but fails to discover the furniture. Our approach, in contrast, is much better at segmenting objects besides the room layout.

密度適応型ネットワークとベースライン手法[20]を比較し、密度適応型ネットワークが有効であることを示す。これらは、我々の密度適応型レイヤー設計の有効性を証明するものである。

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].

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.

**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.

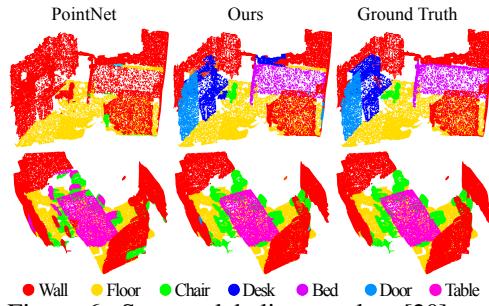


Figure 6: Scannet labeling results. [20] captures the overall layout of the room correctly but fails to discover the furniture. Our approach, in contrast, is much better at segmenting objects besides the room layout.



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

	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	<b>96.09</b>

表 3: SHREC15 非剛体形状分類の例

#### 4.4 Feature Visualization.

図8は、我々の階層型ネットワークの第一階層カーネルで学習された内容を可視化したものである。空間内にボクセルグリッドを作成し、グリッドセル内で特定のニューロンを最も活性化させる局所点セットを集約した（最高100例を使用）。投票数の多いグリッドセルは保持され、ニューロンが認識するパターンを表す3次元点群に再変換される。このモデルは、主に家具から構成される ModelNet40 で学習されるため、平面、二重平面、線、角などの構造が視覚化される。

### 5 Related Work

階層的特徴学習という考え方は、非常に成功している。学習モデルの中でも、畳み込みニューラルネットワーク [10, 25, 8] は最も著名なものの一つである。しかし、畳み込みは、我々の研究の焦点である距離メトリックを持つ非順序点集合には適用されない。

いくつかの非常に最近の研究[20, 28]は、深層学習を非順序集合に適用する方法について研究している。彼らは、点集合が距離メトリックを持っていたとしても、その基礎となる距離メトリックを無視する。その結果、点のローカルコンテキストを捉えることができず、グローバルな集合の変換と正規化に対して敏感である。本研究では、メトリック空間からサンプリングされた点を対象とし、設計において基礎となる距離メトリックを明示的に考慮することで、これらの問題に対処している。

計量空間からサンプリングされた点は、通常、ノイズが多く、サンプリング密度も一様でない。このことは、効率的な点特徴抽出に影響を与え、学習が困難になる原因となる。また、点特徴量の設計において、適切なスケールを選択することが重要な課題である。この問題に関しては、幾何学処理分野や写真測量・リモートセンシング分野において、これまでにいくつかのアプローチが開発されている[19, 17, 2, 6, 7, 30]。本アプローチは、これらのアプローチとは対照的に、エンドツーエンドで点特徴量の抽出と複数の特徴量のスケールのバランスを学習する。

3 次元メトリック空間では、点集合以外に、ボリュームグリッド[21, 22, 29]、幾何グラフ[3, 15, 33]など、深層学習で人気のある表現がいくつかある。しかし、これらの著作では、サンプリング密度が不均一であるという問題が明示的に考慮されていない。

### 6 Conclusion

本論文では、メトリック空間にサンプリングされた点集合を処理するための強力な ニューラルネットワークアーキテクチャであるPointNet++を提案する。PointNet++は、入力点集合のネストされた分割に対して再帰的に閾値化し、距離メトリックに関する階層的な特徴を学習するのに有効である。また、点集合のサンプリングが一様でない問題を解決するために、局所的な点密度に応じてマルチスケール情報をインテリジェントに集約する2つの新しい集合抽象化レイヤーを提案する。これらの貢献により、3次元点群の困難なベンチマークにおいて、最先端の性能を達成することができる。

今後、特に MSG と MRG 層の計算を局所領域ごとに分担することで、本提案ネットワークの推論速度を高速化する方法を考える価値があると思われる。また、高次元メトリック空間において、CNN を用いた手法では計算量が不足するような場合でも、本手法では十分なスケールアップが可能であるため、応用が期待される。

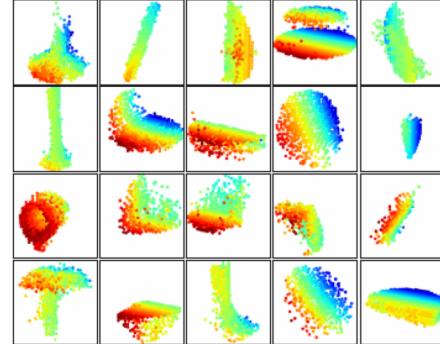


図 8: 第 1 層のカーネルから学習された 3 次元点群パターン。ModelNet40 の形状分類のために学習されたモデル（128 カーネルのうち 20 カーネルがランダムに選択される）。色は点の深さを表す（赤は近い、青は遠い）。

	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	<b>96.09</b>

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.

### 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.

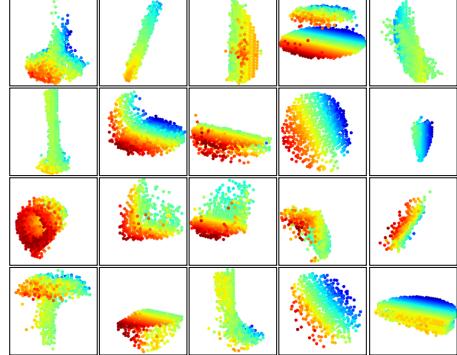


Figure 8: 3D point cloud patterns learned from the first layer kernels. The model is trained for ModelNet40 shape classification (20 out of the 128 kernels are randomly selected). Color indicates point depth (red is near, blue is far).

## References

- [1] M. Aubry, U. Schlickewei, and D. Cremers. 波動カーネル署名. 形状解析のための量子力学的アプローチ. また、このような場合にも、「曖昧さ」の解消が必要である. IEEE, 2011. [2] D. Belton and D. D. Lichti. 2] D. Belton and D. D. Lichti. Classification and segmentation of terrestrial laser scanner point clouds using local variance information. このような場合、そのような情報を利用することができます. [3] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. また、このような場合、「李鍊學」は、「李鍊學」と「李鍊學」と「李鍊學」の中間的な存在となる. [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 (情報量の多い3Dモデルリポジトリ). テクニカルレポート 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. 3d ライダ一点群における次元に基づくスケール選択. また、このような場合にも、そのようなデータを利用することができます. [7] A. Gressin, C. Mallet, J. Demantké, and N. David. また、このような場合にも、そのような情報を利用することができます. また、このような場合にも、そのようなデータを利用することができます. [8] K. He, X. Zhang, S. Ren, and J. Sun. 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 と J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. このような場合、「曖昧さ」を解消することが重要である. また、このような場合、「曖昧さ」を解消するために、「曖昧さ」と「曖昧さ」の間の「曖昧さ」の間の「曖昧さ」を解消する必要がある. [11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 勾配に基づく学習の文書認識への応用. また、このような勾配を利用することで、勾配を利用した学習が可能となる. [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. R. 中西 宙、A. P. Neto, L. G. Nonato、大渕 理恵、K. Pevzner, D. Pickup, P. Rosin, A. Sharf, L. Sun, X. Sun, S. Tari, G. Unal、および R. C. Wilson. 非剛体3次元形状検索. I. Pratikakis, M. Spagnuolo, T. Theoharis, L. V. Gool, and R. Veltkamp, editors, Eurographics Workshop on 3D Object Retrieval. にて. 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. このような場合、「geodesic moments」を用いたディープラーニングは、3次元の形状分類に有効である. 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 two-manifolds. (三角形2多様体のための離散微分幾何演算子). 可視化と数学, 3(2):52–58, 2002. [17] N. J. MITRA, A. NGUYEN, and L. GUIBAS. また、このような場合にも、そのような問題を解決するために、より効果的な方法を検討する必要がある. [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. また、このような場合にも、そのような技術的背景を考慮する必要がある. [20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: 3d分類とセグメンテーションのための点集合のディープラーニング. [21] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. 3d データ上のオブジェクト分類のためのボリュームとマルチビューCNN. というものである. Computer Vision and Pattern Recognition (CVPR), IEEE, 2016. [22] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: 高解像度で深い3d表現を学習する. arXiv preprint arXiv:1611.05009, 2016. [23] R. M. Rustamov, Y. Lipman, and T. Funkhouser. バリセントリック座標を用いた内部距離. Computer Graphics Forum, volume 28, pages 1279–1288 にて. Wiley Online Library, 2009. [24] P. Y. Simard, D. Steinkraus, and J. C. Platt. ICDAR, volume 3, pages 958–962, 2003. [25] K. Simonyan and A. Zisserman. 大規模画像認識のための非常に深い畳み込みネットワーク. arXiv preprint arXiv:1409.1556, 2014. [26] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. 3d 形状認識のためのマルチビュー畳み込みニューラルネットワーク. 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. 熱拡散に基づく簡潔で証明可能な情報量の多いマルチスケールシグネチャ. また、このような場合にも、「俯瞰的な視点」を持つことが重要である. Wiley Online Library, 2009. [28] O. Vinyals, S. Bengio, and M. Kudlur. 順序は重要です. Sequence to sequence for sets. arXiv preprint arXiv:1511.06391, 2015.

## 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. Lichten. 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. Ulusoy, 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. optimal neighborhoods, relevant features and efficient classifiers に基づく意味的点群解釈。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: 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 (3次元形状コレクションにおける領域アノテーションのためのスケーラブルなアクティブフレームワーク)。SIGGRAPH Asia, 2016. [33] L. Yi, H. Su, X. Guo, and L. Guibas. Syncspecnn: 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. Syncspecnn: Synchronized spectral cnn for 3d shape segmentation. *arXiv preprint arXiv:1612.00606*, 2016.

## Supplementary

### A Overview

この補足資料は、本論文の実験の詳細を提供し、提案手法を検証・分析するための実験をより多く含んでいます。

Sec B では、本論文の実験に使用した具体的なネットワークアーキテクチャを提供し、データ準備とトレーニングにおける詳細についても説明する。また、Sec C では、部品分割や近傍探索に関するベンチマーク性能、サンプリングのランダム性に対する感度、時間空間の複雑性など、より多くの実験結果を示している。

### B Details in Experiments

アーキテクチャプロトコル 本ネットワークアーキテクチャを説明するために、以下の表記を用いる。

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

$FC(1, dp)$  は、幅1、ドロップアウト率 $dp$ の完全連結層を表す。 $FP(1, 1, \dots, 1, d)$  は、 $d$ 個の完全連結層を持つ特徴伝搬(FP)レベルである。これは、補間とスキップリンクから連結された特徴を更新するために使用される。最後のスコア予測層を除き、全ての完全連結層に一括正規化、ReLUが続く。

#### B.1 Network Architectures

全ての分類実験において、我々は以下のアーキテクチャ (Ours SSG) を異なるKを使用している。  
(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:

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

'クロスレベル・マルチレゾリューション・グルーピング (MRG) ネットワークのアーキテクチャは、3つのブランチを使用しています。

- 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:  $SA(256, 512, 1024)$ .

分岐1と分岐2は連結され、分岐4へ送られる。分岐3と分岐4の出力は連結され、 $FC(512, 0.5) \rightarrow FC(256, 0.5) \rightarrow FC(K)$ に送られ、分類されます。

意味的シーナラベリング用ネットワーク (FPの最後の2つの完全接続層は、ドロップアウトが続く layers with drop ratio 0.5):

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

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

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

## 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.

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

$FC(l, dp)$  represents a fully connected layer with width  $l$  and dropout ratio  $dp$ .  $FP(l_1, \dots, l_d)$  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):

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

## B.2 Virtual Scan Generation

本節では、ScanNetのシーンから、サンプリング密度が一様でないラベル付き仮想スキャンを生成する方法について説明する。ScanNetの各シーンに対して、カメラ位置を床面のセントロイドから1.5m上に設定し、カメラの向きを水平面内で8方向に均等に回転させる。各方向には、 $100\text{px} \times 75\text{px}$ の画像平面を用い、カメラから各画素を通る光線をシーンに投影する。これにより、シーン内の可視点を選択する方法が得られます。このようにして、各テストシーンに対して8つの仮想スキャンを生成することができました。その例を図9に示します。点サンプルは、カメラに近い領域でより密になっていることに注意してください。

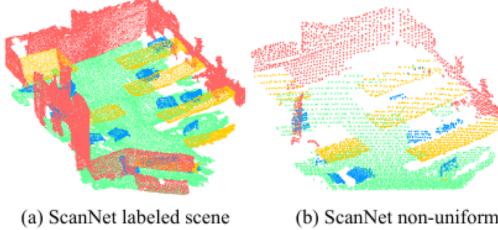


図 9: ScanNet から生成された仮想スキャン

## B.3 MNIST と ModelNet40 の実験詳細

MNISTの画像では、まず、全ての画素の強度を[0, 1]の範囲に正規化し、0.5以上の強度を持つ画素を有効な数字画素として選択する。次に、画像中の数字ピクセルを、画像中心を原点とする[-1, 1]の座標を持つ2次元点群に変換する。拡張点は、点群を一定の基数（ここでは512）に追加するために作成される。初期点群にジッターを与え（ガウス分布 $N(0, 0.01)$ のランダムな移動で、0.03にクリッピング）、拡張点を生成する。ModelNet40では、CADモデルの表面から、顔の面積に基づいてN個の点を一様にサンプリングする。

全ての実験において、学習率0.001のAdam[9]オプティマイザーを使用し、学習を行った。データ補強のため、物体をランダムに拡大縮小し、物体の位置と点サンプルの位置を揺動させる。また、ModelNet40のデータ拡張のために、[21]に従ってオブジェクトをランダムに回転させる。学習にはTensorFlowとGTX 1080, Titan Xを使用しています。全てのレイヤーをCUDAで実装し、GPUで動作させる。また、このモデルを収束まで学習させるのに20時間程度かかっている。

## B.4 ScanNet Experiment Details

ScanNetシーンから学習データを生成するために、初期シーンから $1.5\text{m} \times 1.5\text{m} \times 3\text{m}$ のキューブをサンプリングし、ボクセルの $\geq 2\%$ が占有され、表面のボクセルの $\geq 70\%$ が有効なアノテーションを持っているキューブを保持します（これは、[5]と同じセットアップです）。このようなトレーニングキューブをオンザフライでサンプリングし、右上軸に沿ってランダムに回転させる。拡張された点は、一定の基数（我々の場合は8192）になるように点セットに追加される。テスト時には、同様にテストシーンを小さなキューブに分割し、まずキューブ内の全ての点のラベル予測を行い、次に同じシーンからの全てのキューブのラベル予測を統合する。もし、ある点が異なるキューブから異なるラベルを得た場合、最終的な点のラベル予測を得るために、多数決を行うだけである。

## B.5 SHREC15 Experiment Details

学習用、テスト用とともに、各シェイプ上の1024点をランダムにサンプリングする。入力の固有特徴を生成するために、100次元のWKS、HKS、マルチスケールガウス曲率をそれぞれ抽出し、各点に対して300次元の特徴ベクトルを導く。次に、PCAを行い、特徴量を64次元に削減する。我々は、点近傍を選択しながら、我々の非ユークリッドメトリック空間を記述するため使用される測地線距離を模倣するために、[23]に従って8次元のエンベッドを使用する。

## 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.

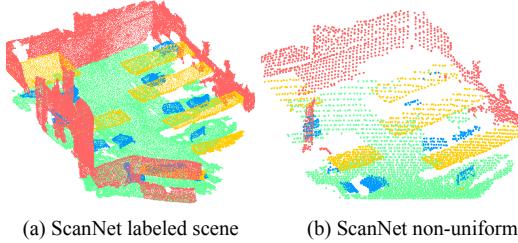


Figure 9: Virtual scan generated from ScanNet

## 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 TensorFlow 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.

## C More Experiments

本節では、提案するネットワークアーキテクチャを検証・分析するために、より多くの実験結果を提供する。

### C.1 Semantic Part Segmentation

32]の設定に従い、各形状に対するカテゴリラベルが既知であると仮定して、部品分割タスクに対する我々のアプローチを評価する。点群によって表現された形状を入力とし、各点に対応する部品ラベルを予測するタスクである。データセットには16クラスからなる16,881個の形状が含まれ、合計で50個の部品が注釈されている。我々は[4]に従い、公式の訓練テスト分割を使用する。

各点にその法線方向を装備することで、基本的な形状をより良く表現することができる。このようにすることで、[32, 33]で用いられているような、手作業で作られた幾何学的特徴を取り除くことができる。我々は、表4において、我々のフレームワークを従来の学習ベースの技術[32]、および、最先端の深層学習アプローチ[20, 33]と比較している。評価指標としては、全パートクラスで平均化したIoU (Point Intersection Over Union) を使用する。クロスエントロピーの損失は、学習中に最小化される。平均すると、我々のアプローチが最も良い性能を達成した。20]と比較すると、我々のアプローチはほとんどどのカテゴリで良好な結果を得ており、詳細な意味理解のための階層的特徴学習の重要性が証明された。我々のアプローチは、異なるスケールで近接グラフを暗黙的に構築し、これらのグラフ上で操作しているとみなすことができ、したがって、[33]のようなグラフCNNアプローチと関連していることに注目されたい。我々のマルチスケール近傍選択の柔軟性と集合演算ユニットのパワーのおかげで、我々は[33]と比較してより良い性能を達成することができた。また、[33]とは異なり、高価な固有値分解を行う必要がない。これらのことから、本アプローチは大規模点群解析により適している。

	mean	aero	bag	cap	car	椅子	耳	ギター	ナイフ	ランプ	ラップトップ	モーター	マグカップ	ピストル	ロケット
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
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
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
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

表 4: ShapeNet パートデータセットにおけるセグメンテーション結果。

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

ここでは、局所近傍を選択するための2つのオプションを比較する。本論文では、半径に基づくボールクエリを使用した。この実験では、全ての学習とテストは一様なサンプリング密度を持つ ModelNet40 形状で行われます。1024点が使用された。表5からわかるように、半径ベースのボールクエリはkNNベースの方法よりわずかに優れています。しかし、非常に非均一な点群では、kNNベースのクエリはより悪い汎化能力をもたらすと推測される。また、半径を少し大きくすることで、局所的なパターンをより豊かに捉えることができるため、性能に役立つことがわかる。

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

表5：近傍探索の効果。評価指標はModelNet 40テストセットでの分類精度(%)である。

### C.3 最遠点サンプリングにおけるランダム性の効果。

集合抽象化レベルのサンプリング層では、点集合のサブサンプリングにFarthest Point Sampling (FPS) を用いている。しかし、FPSのアルゴリズムはランダムであり、サブサンプリングはどの点が最初に選択されるかに依存する。ここでは、このランダム性に対する我々のモデルの感度を評価する。表6では、ModelNet40で学習させたモデルの特徴安定性と分類安定性をテストしている。

## 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	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate	table	board
							phone											
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 result 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.

特徴の安定性を評価するために、全てのテストサンプルについて、異なるランダムシードで10回大規模な特徴を抽出します。そして、10回のサンプリングにおける各形状の特微量の平均値を算出する。次に、各特徴の平均値からの差の標準偏差を計算する。最後に、すべての特微量の標準偏差を平均し、表に示す。特微量は0から1に正規化されているので、0.021の差は、特微量の標準偏差が2.1%であることを意味する。

分類については、ModelNet40の全てのテスト形状において、テスト精度の標準偏差が0.17%しかなく、サンプリングのランダム性にロバストであることが観測された。

Feature difference std.	Accuracy std.
0.021	0.0017

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

#### C.4 Time and Space Complexity.

表7は、いくつかの点群ベースの深層学習手法の間の時間と空間のコストの比較をまとめたものである。GTX 1080一台でTensorFlow 1.1を用いてバッチサイズ8で前方時間を記録している。最初のバッチはGPUの準備があるため、無視しています。PointNet (vanilla) [20]が最も時間効率が良いが、密度適応層を用いない我々のモデルは最小のモデルサイズとそれなりの速度を達成した。

MSGは、非一様サンプリングデータで良好な性能を示すが、マルチスケール領域特徴抽出のため、SSG版に比べ2倍のコストがかかるることは注目に値する。MRGはレイヤーをまたいで領域を使用するため、MSGと比較してより効率的である。

	PointNet (vanilla)	PointNet	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

表 7: いくつかのネットワークのモデルサイズと推論時間(フォワードパス)。

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)	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.