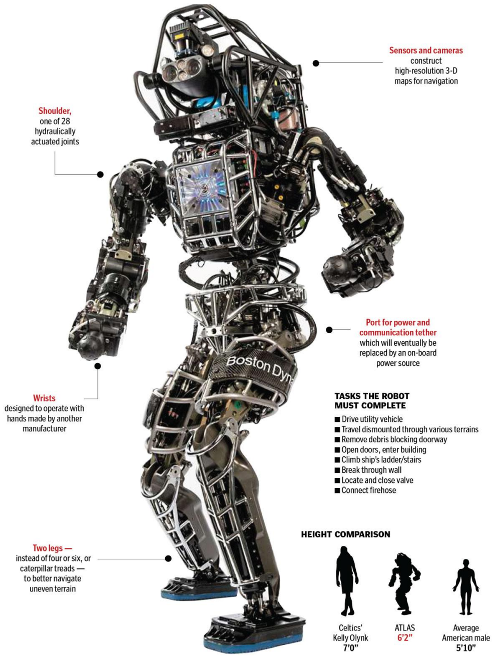
PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space

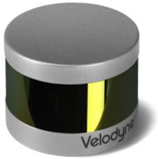
Presenter: Lance Zhang

08/31/2023

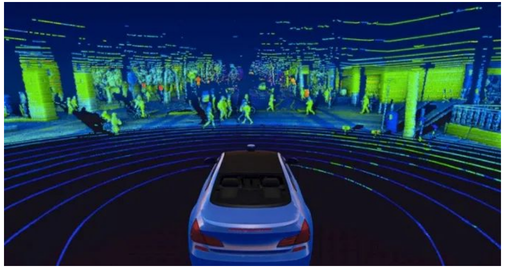
Motivation and Main Problem



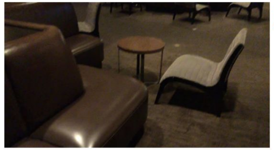
It is important to effectively process **point cloud** data!

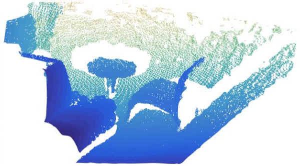


LiDAR Sensors



Depth Cameras



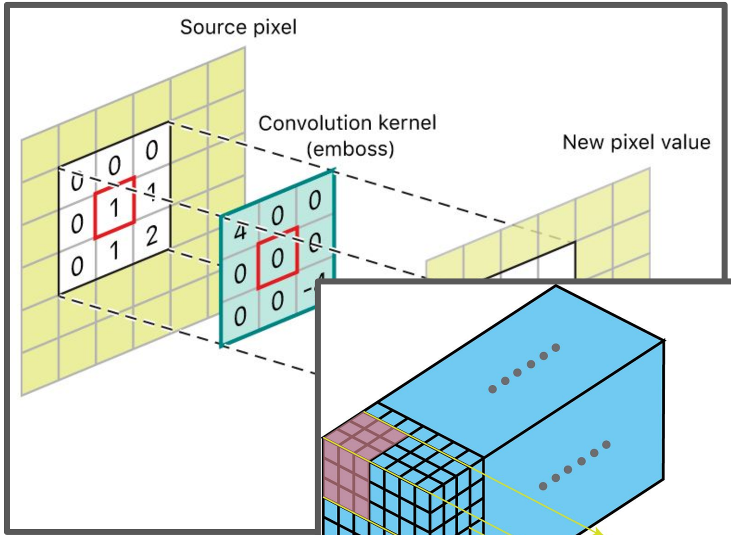


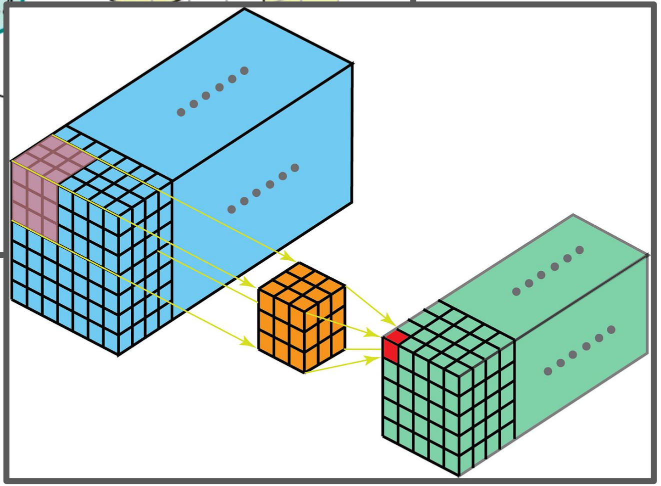
Motivation and Main Problem

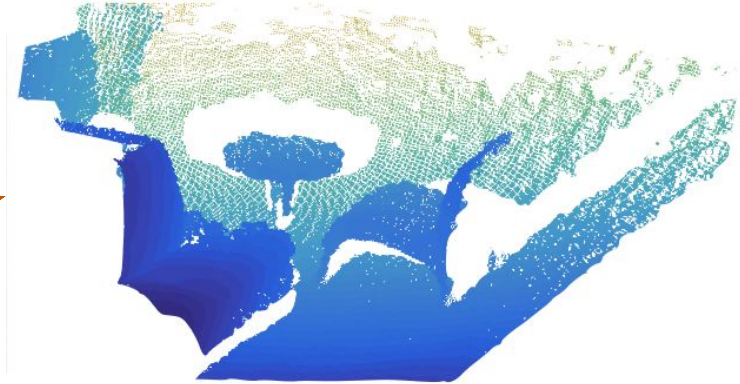
However, few works have studied deep learning for point sets and leveraged their **unique characteristics**:

- Irregular data structure (lack of fixed grid)
- Density variability
- Importance of both local & global contexts
- ...

Motivation and Main Problem







We need a algorithm that allows:

1. Hierarchical feature learning
2. Point cloud processing without grid-based representation

Problem Setting

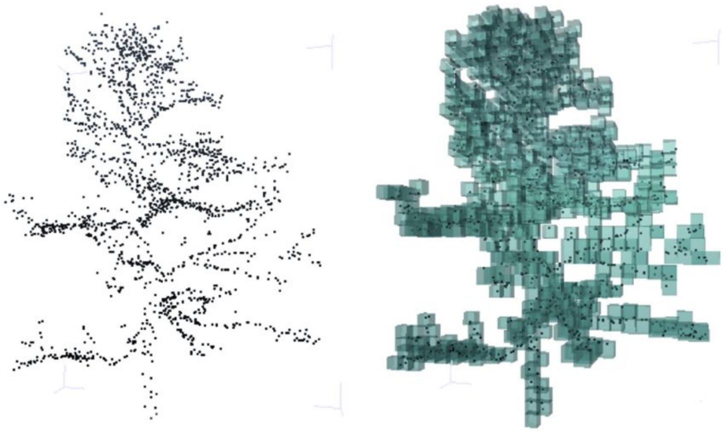
$$\mathcal{X} = (M, d)$$

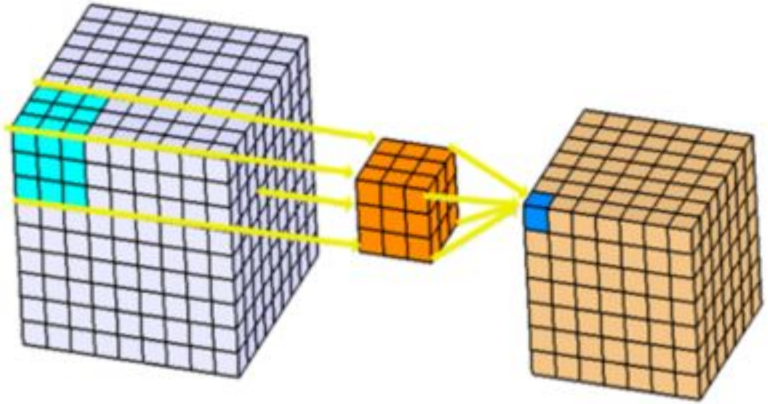
$M \subseteq \mathbb{R}^n$: a set of points in n-dimensional space

d: Euclidean distance metric

Objective: Learning set functions f that takes the metric space χ as input and produce information of semantic interest (for classification & segmentation)

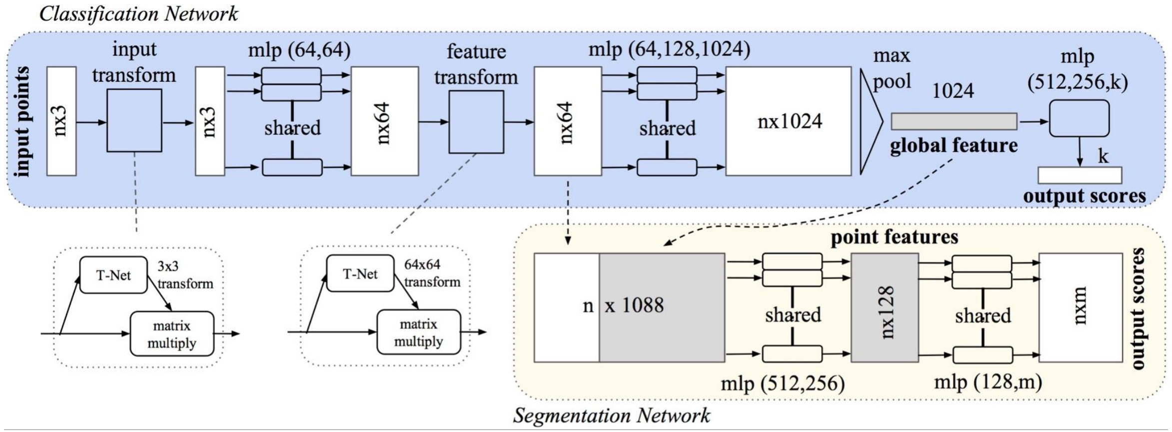
Related Work: Voxelization





- ✅ Can apply 3D CNN to point cloud data
- ❌ Very sparse mesh (most of 3D space is empty)

Prior Work: PointNet



- ✅ Directly processing point cloud data, permutation invariant
- ❌ Does not take into account density variability
- ❌ Does not capture both global and **local** features

Proposed Approach

PointNet++ takes a **hierarchical approach** to feature learning by creating "neighborhoods" of points in various resolutions

i.e. Applying PointNet **recursively** on a nested partitioning of the point set

Set abstraction layer:

1. Sample centroids (center of neighborhoods)
2. Group points (for each neighborhood)
3. PointNet layer (feature extraction for that level)

PointNet++: Layers

Sampling Layer

Iterative Farthest Point Sampling (FPS)

$$\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\} \subset \{x_1, x_2, \ldots, x_n\}$$
$$x_{i_j} \rightarrow \text{the most distant point from the set } \{x_{i_1}, \ldots, x_{i_{j-1}}\}$$

Grouping Layer

Select points for each neighborhood centroid through ball queries

- Number of points can vary significantly

PointNet++: Layers

PointNet Layer

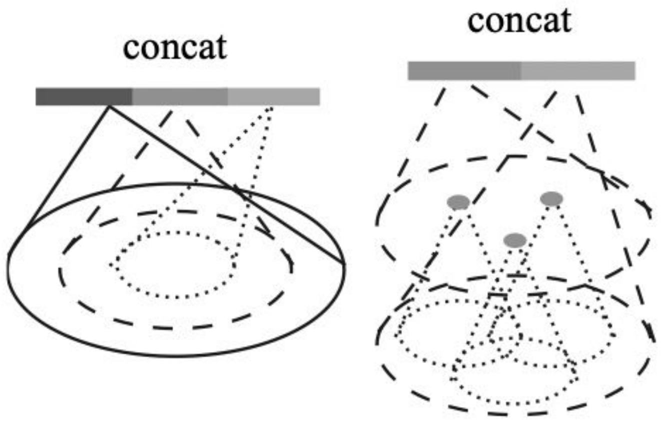
Applies a small PointNet to a given set of points for feature extraction

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

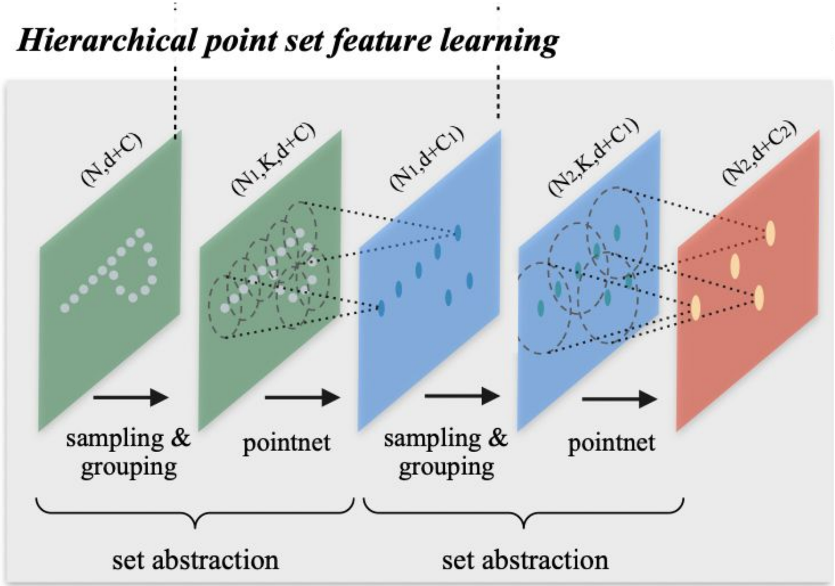
Addressing **non-uniform sampling density**:

- Multi-scale grouping (MSG, left)
- Multi-resolution grouping (MRG, right)
- Random input dropout

$$\theta \sim U[0, p] \text{ (e.g., } p = 0.95)$$



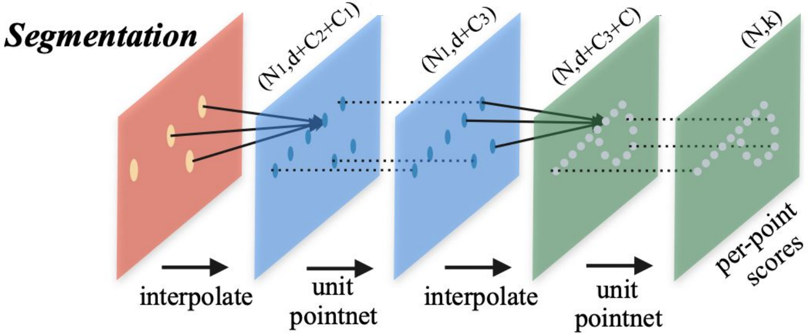
PointNet++: Putting It All Together



Propagation for Segmentation

We need to classify each point to perform segmentation ⇒ propagate centroid features to original point set:

Inverse Distance Weighted Interpolation



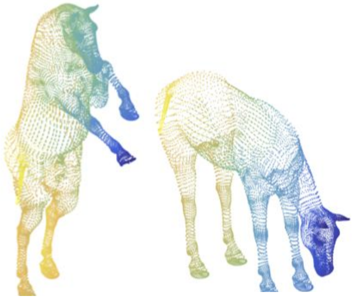
$$f^{(j)}(x) = \frac{\sum_{i=1}^{k} w_i(x) f_i^{(j)}}{\sum_{i=1}^{k} w_i(x)} \quad \text{where} \quad w_i(x) = \frac{1}{d(x, x_i)^p},\ j = 1, \dots, C$$

Experimental Setup: Datasets

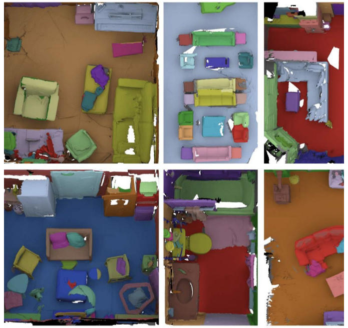
PointNet++ was evaluated on four datasets in **various domains**:

MNIST
(2D)

ModelNet40
(3D models)



SHREC15
(non-rigid 3D models)



ScanNet
(3D Indoor Scenes)

Complexity

Experimental Setup: Algorithms

PointNet++

SSG (Single-Scaled Grouping)

SSG + DP (with input dropout)

MSG + DP (best but expensive)

MRG + DP

Baselines (3D)

Subvolume (volumetric CNN)

MVCNN (multi-view CNN)

PointNet

PointNet (Vanilla)

Baselines (2D, MNIST)

MLP

LeNet5

Network in Network

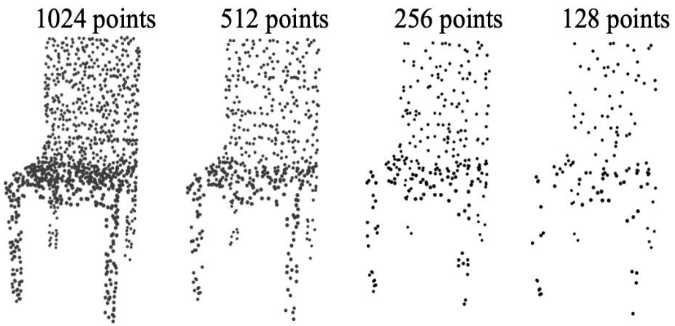
Experimental Results

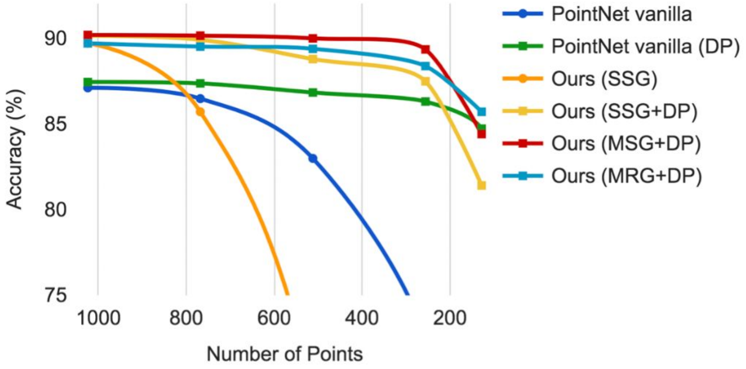
Method	Error rate (%)
Multi-layer perceptron [24]	1.60
LeNet5 [11]	0.80
Network in Network [13]	**0.47**
PointNet (vanilla) [20]	1.30
PointNet [20]	0.78
Ours	0.51

Table 1: MNIST digit classification.

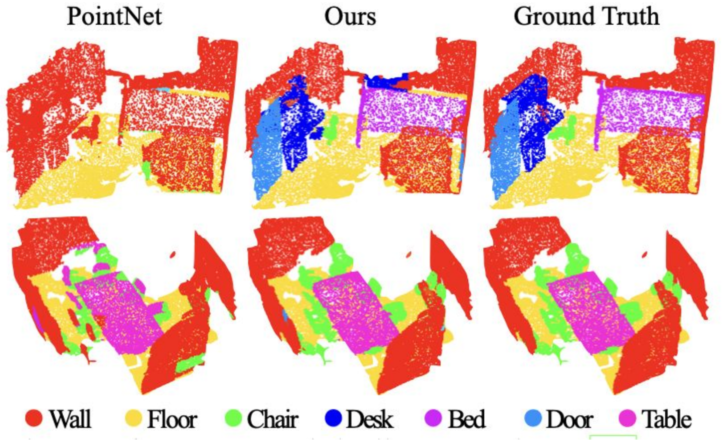
Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9**

Table 2: ModelNet40 shape classification.





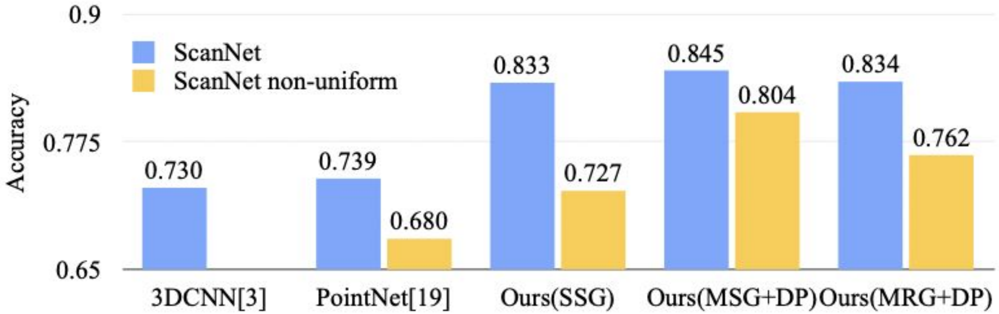
Experimental Results



ScanNet

- Complex scenes
- Closest to real-world sensor data

Significant improvements over 3DCNN and PointNet, affirming the importance of hierarchical feature learning



Experimental Results

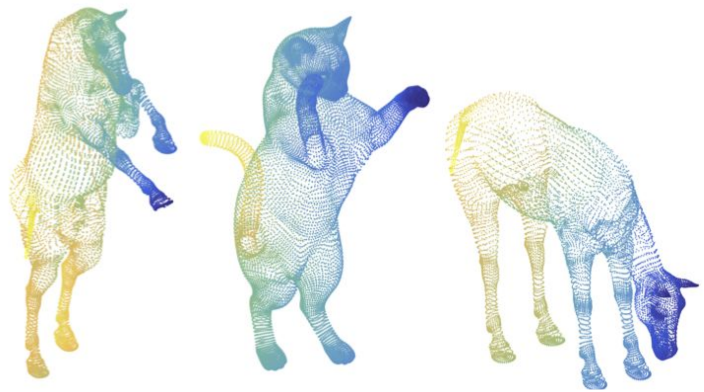
	Metric space	Input feature	Accuracy (%)
DeepGM [14]	-	Intrinsic features	93.03
Ours	Euclidean	XYZ	60.18
	Euclidean	Intrinsic features	94.49
	Non-Euclidean	Intrinsic features	**96.09**

Table 3: SHREC15 Non-rigid shape classification.

SHREC15 models are 2D surfaces embedded in 3D

Learning from:

- Non-Euclidean (geodesic) metric space
- Intrinsic features (WKS, HKS, and multi-scale Gaussian curvature)



Discussion of Results

The PointNet++ architecture effectively learns **multi-scale feature representations** from **point sets** sampled from a metric space.

- The architecture achieved SOTA performance on a variety of datasets
- Ablations also showed the algorithm:
  - Is robust to density variations
  - Benefits from combining features from different scales
  - Can learn structure from different metric spaces/input features

Performance improvements are more significant for complex scenes than 2D data and simple 3D models

Limitations

The MSG (and MRG) layers are expensive in computation and memory

- Local PointNet at every centroid
- Large amounts of centroids at lower levels

➔ **Accelerating inference by sharing computations in local regions**

Lack of geometric awareness

- Considers neighborhoods but not the geometric structure

➔ **Incorporate such information into the algorithm**

Future Work

- **Scalability and Efficiency** (as mentioned)
- **Fine-Grained Geometry:** consider geometric features (curvatures, edges, corneers) in addition to the distance metric
- **Attention Mechanisms:** allow the model to dynamically focus on informative parts of the point cloud, instead of relying on max pooling
- **Dynamic Scenes:** incorporate temporal context, better suited for real-time sensor data

Extended Readings

- **Dynamic Graph CNN for Learning on Point Clouds**: An alternative approach for processing point clouds using CNNs, representing them through graphs instead of voxelization
- **Point Transformer**: Self-attention layers for point clouds
- **Deep Learning for 3D Point Clouds**
- **PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation**: foundation for PointNet++

Summary

- PointNet++ addresses the challenge of processing 3D point cloud data for tasks such as model classification and scene segmentation
- Point clouds lack the regular grid structure and often have non-uniform density
- Prior work does not account for the **unique properties** of point clouds and fail to capture both local and global structures simultaneously
- By recursively applying PointNet on nested partitionings of the original data, PointNet++ allows for **multi-scale feature extraction**
- The proposed algorithm achieved **SOTA performance** on a variety of datasets, while also being robust to non-uniform density and different metric spaces

Thank you!!