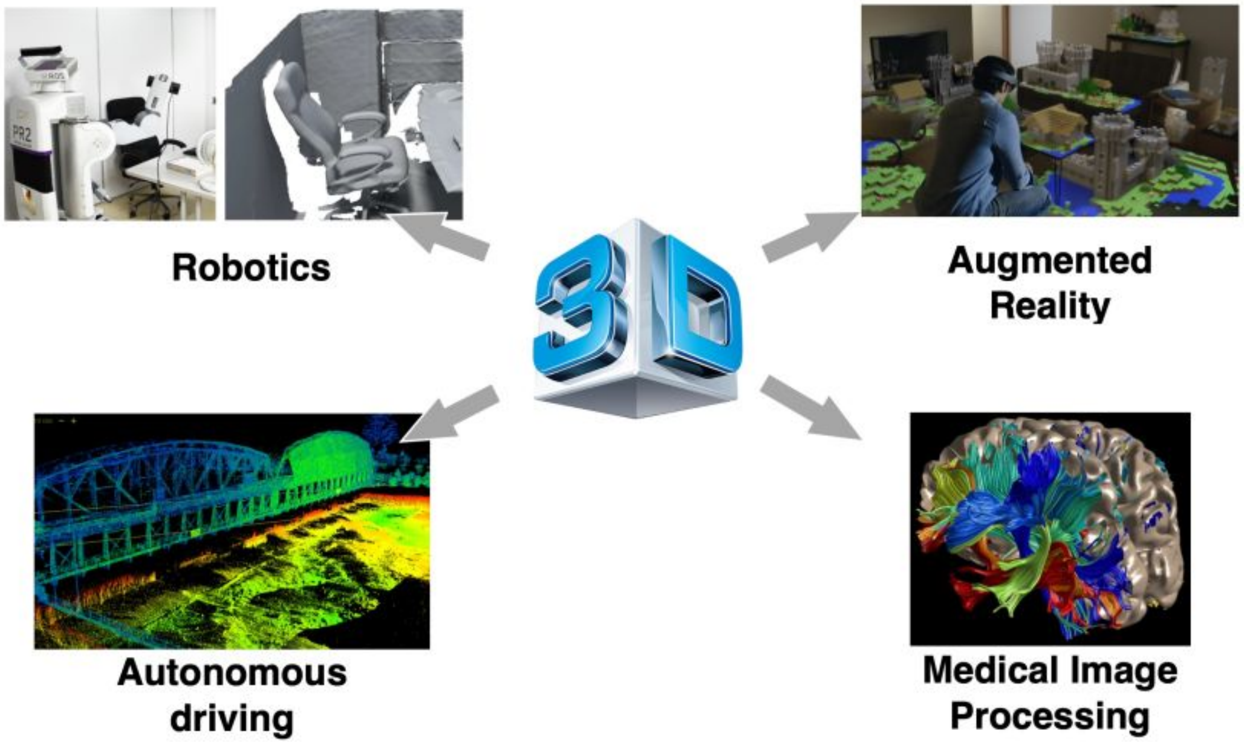
3D Vision

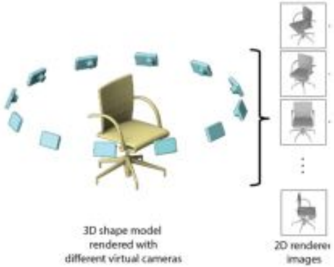
Representations and Learning

Akash Kumbar
akash.kumbar@research.iiit.ac.in
(4th June 2025)

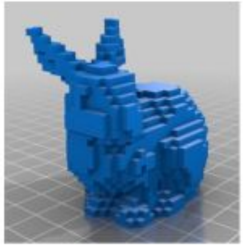
Applications of 3D data



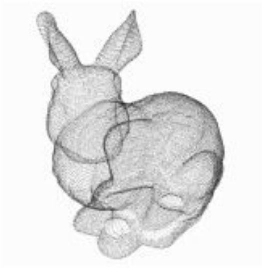
3D data representations



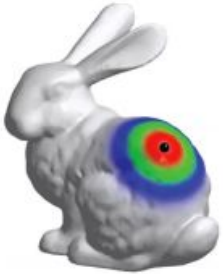
Multi-view



Volumetric

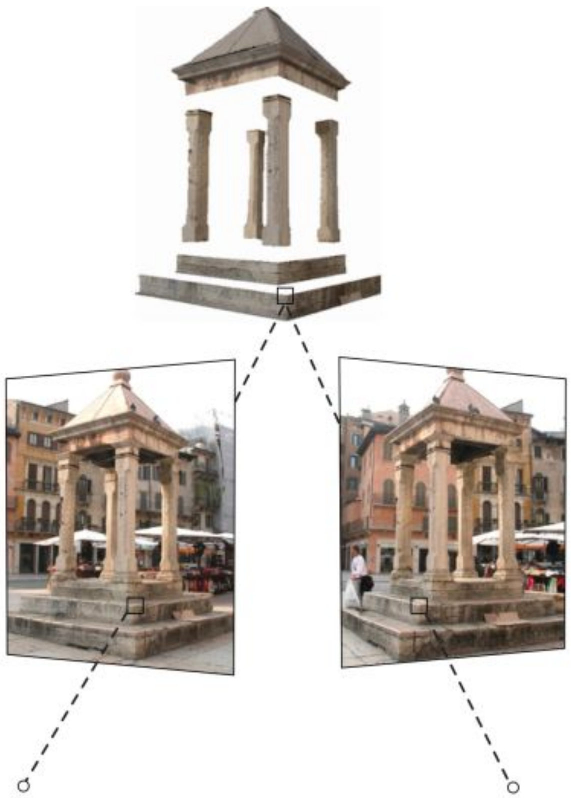


Point Cloud

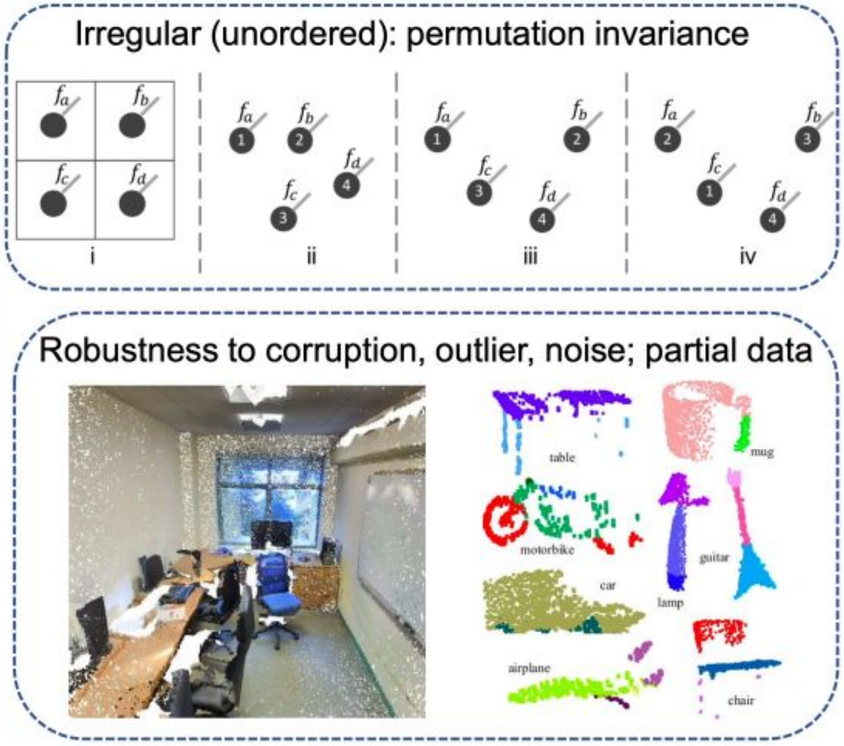


Mesh (Graph CNN)

Traditional 3D Vision: Multi-view geometry

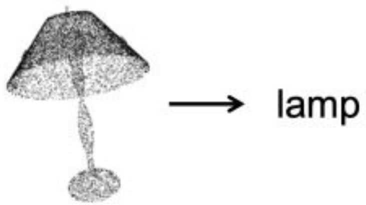


Challenges with Point Cloud

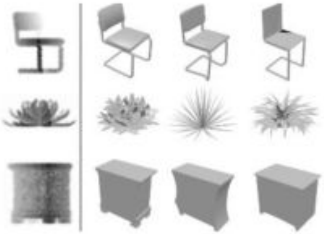


- A point cloud consists a set of points:
    - Unstructured
    - Irregularly distributed in 3D space
    - Unordered
- While ConvNets are great for images, they are not suitable for for point clouds
- Deep learning requires a large amount of data, but annotating point cloud is challenging.

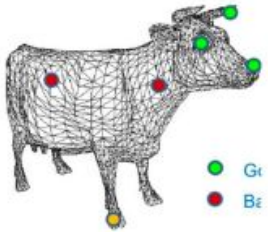
Tasks



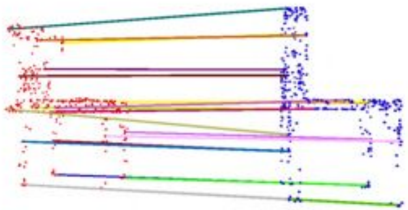
shape classification



shape retrieval

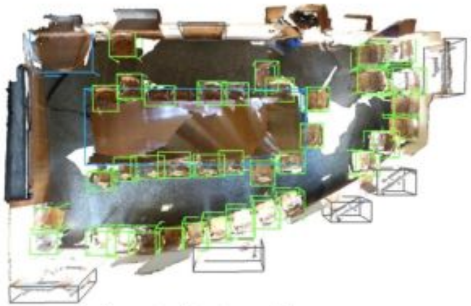


keypoint detection

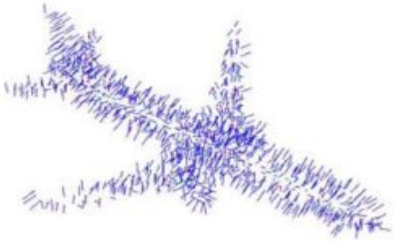


shape correspondence

semantic segmentation



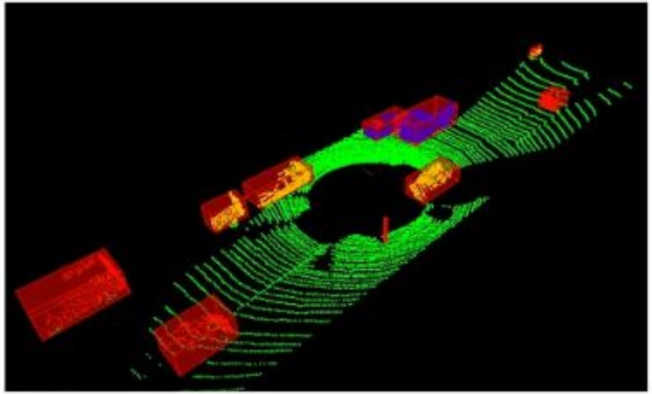
object detection



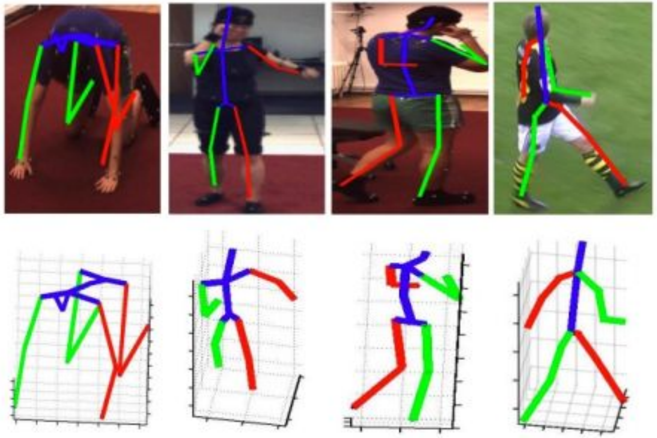
normal estimation

......

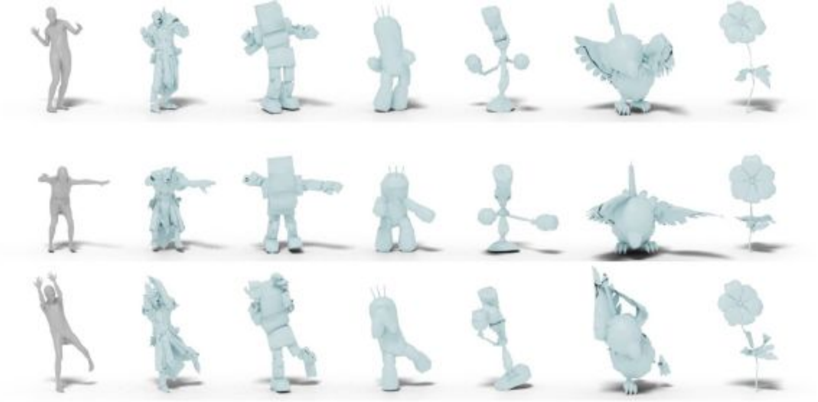
Tasks



Tracking

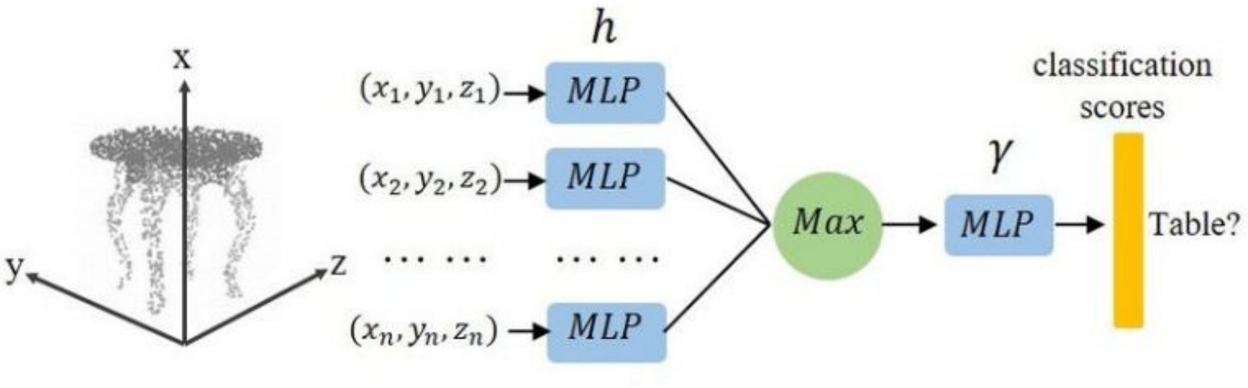


Pose Estimation



Pose Transfer

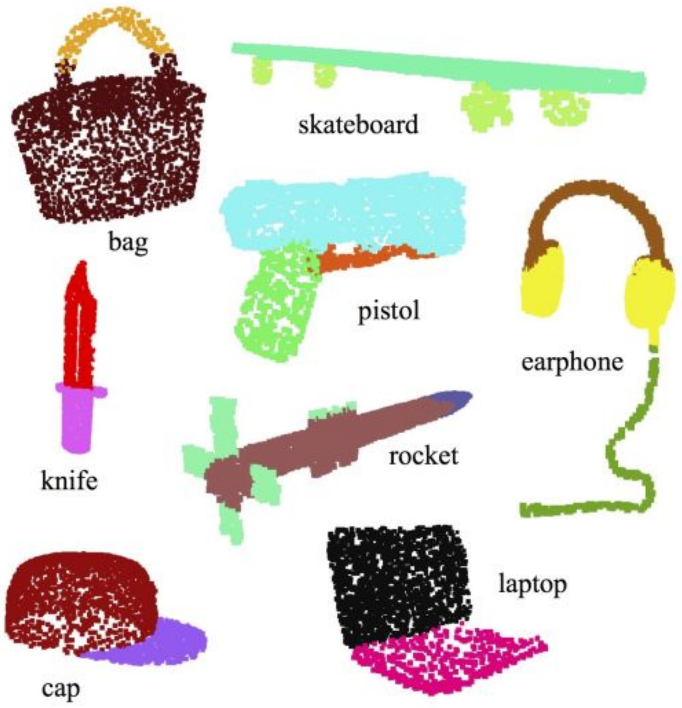
3D object classification



- For the object classification, we consider labeled point cloud (P, y), where P is a point cloud in a collection of point clouds P and y is an integer class label (table, chair, aeroplane) that takes values in the set Y = {1,⋯ ,K}.
- Model outputs K scores for all the K classes.

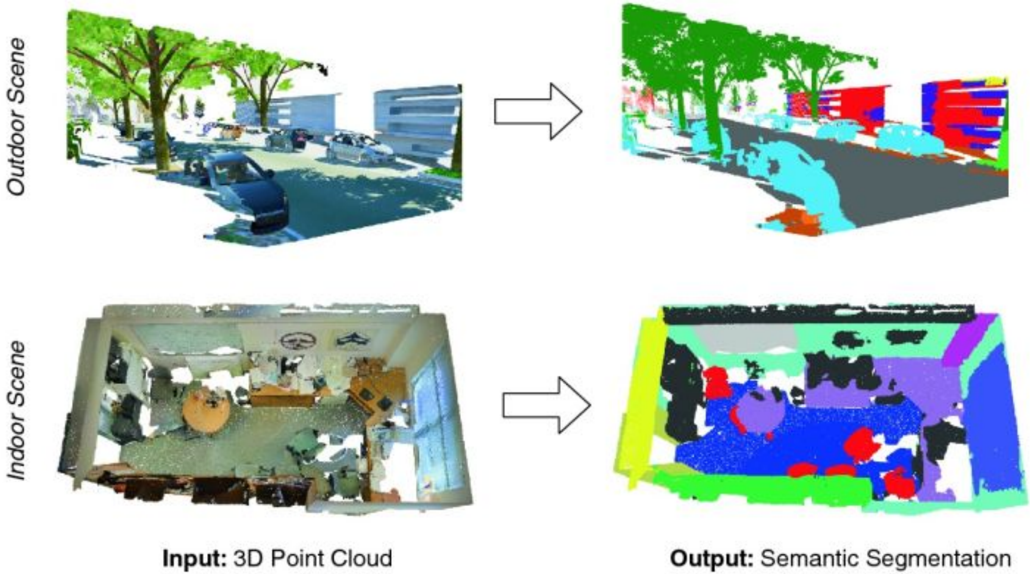
Part Segmentation

For part segmentation, model will output n × m scores for each of the n points and each of the m part categories (wings of the aeroplane, earbuds of the headphone) of objects.



Semantic Segmentation

For semantic segmentation, model outputs n × m scores for each of the n points and each of the m semantic sub-categories(car, tree in outdoor scene)



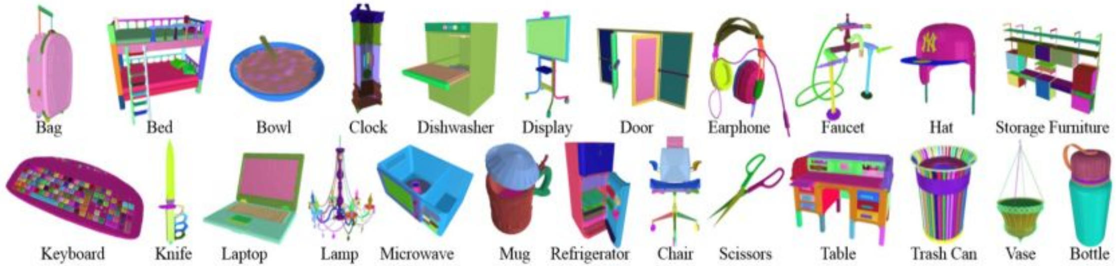
Dataset for 3D Objects: ShapeNet

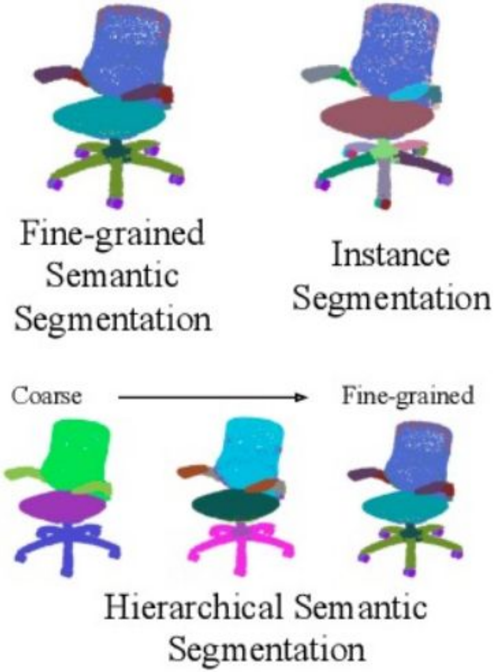
- 3D Object Scans
- Similar to ModelNet
- Used for classification

Chang et al., "**ShapeNet: An Information-Rich 3D Model Repository**" , *arXiv 2015.*

Dataset for 3D Object Parts: ShapeNetPart

- Fine-grained (towards mobility)
- Instance-level
- Hierarchical

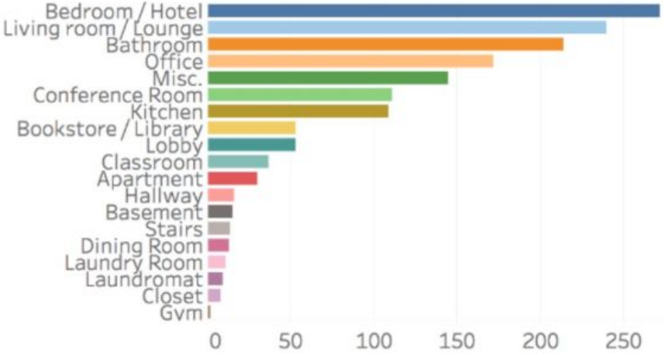




Mo et al., “PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding ”, CVPR 2019

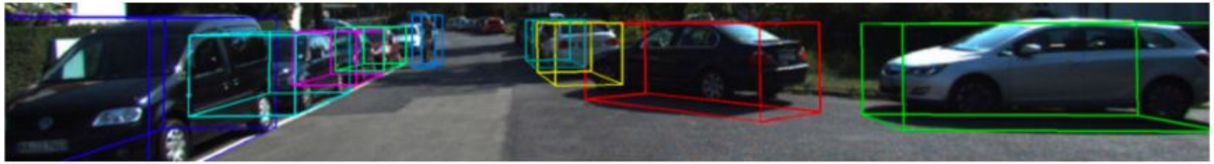
Dataset for 3D Object Parts: ScanNet++

- 1006 3D Indoor Scenes
- Multi-Modal RGB Data
- Comprehensive 3D Reconstructions
- Semantic and Instance-Level Annotations



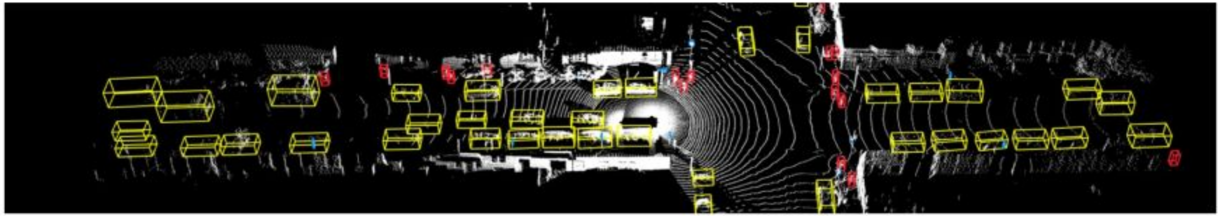
Datasets for Outdoor 3D Scenes

KITTI: LiDAR data, labeled by 3D b.boxes

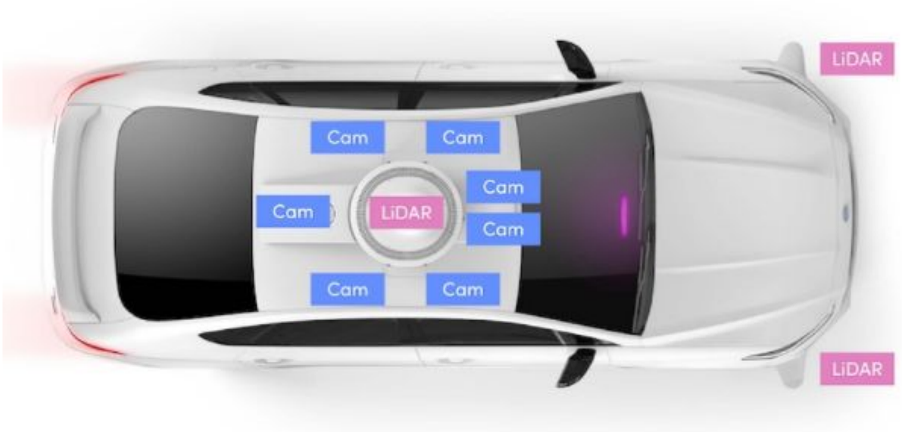


Semantic KITTI: LiDAR data, labeled per point

Waymo Open Dataset: LiDAR data, labeled by 3D b.boxes



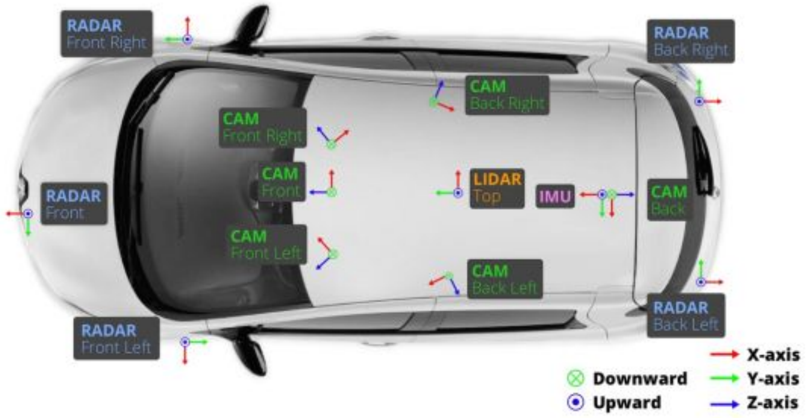
Data Generation - Layouts



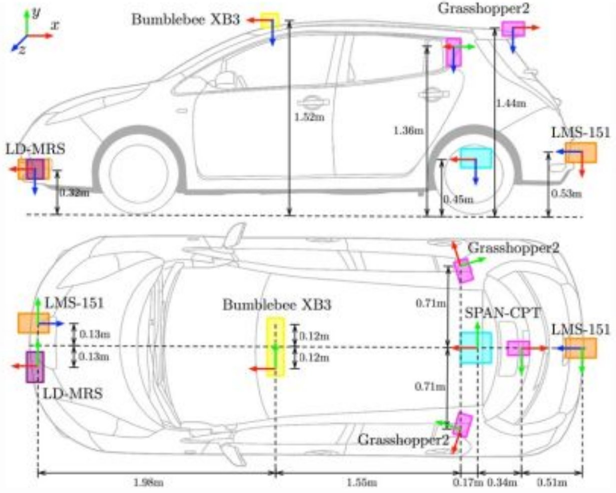
Lyft Level 5 Dataset

Argoverse

Data Generation - Layouts



NuScenes



Sensor positions on vehicle. Coordinate frames use the convention x = forward (red), y = right (green), z = down (blue).

Apolloscape

Publicly available datasets

Datasets\Tasks	Detection	Tracking	Prediction	Segmentation
KITTI [1]	Benchmark	Benchmark		Benchmark
Oxford RobotCar [2]			Benchmark	
Apolloscape [3]			Benchmark	Benchmark
NuScenes [4]	Benchmark	Benchmark	Benchmark	Benchmark
Argoverse [5]	Benchmark	Benchmark	Benchmark	
Lyft Level 5 [6]	Benchmark	Benchmark	Benchmark	
Waymo [7]	Benchmark	Benchmark	Benchmark	

Datasets for 3D Shape Classification							
Name and Reference	Year	#Samples	#Classes	#Training	#Test	Type	Representation
McGill Benchmark [23]	2008	456	19	304	152	Synthetic	Mesh
Sydney Urban Objects [24]	2013	588	14	-	-	Real-World	Point Clouds
ModelNet10 [6]	2015	4899	10	3991	605	Synthetic	Mesh
ModelNet40 [6]	2015	12311	40	9843	2468	Synthetic	Mesh
ShapeNet [8]	2015	51190	55	-	-	Synthetic	Mesh
ScanNet [11]	2017	12283	17	9677	2606	Real-World	RGB-D
ScanObjectNN [7]	2019	2902	15	2321	581	Real-World	Point Clouds
Datasets for 3D Object Detection and Tracking							
Name and Reference	Year	#Scenes	#Classes	#Annotated Frames	#3D Boxes	Secne Type	Sensors
KITTI [14]	2012	22	8	15K	200K	Urban (Driving)	RGB & LiDAR
SUN RGB-D [25]	2015	47	37	5K	65K	Indoor	RGB-D
ScanNetV2 [11]	2018	1.5K	18	-	-	Indoor	RGB-D & Mesh
H3D [26]	2019	160	8	27K	1.1M	Urban (Driving)	RGB & LiDAR
Argoverse [27]	2019	113	15	44K	993K	Urban (Driving)	RGB & LiDAR
Lyft L5 [28]	2019	366	9	46K	1.3M	Urban (Driving)	RGB & LiDAR
A*3D [29]	2019	-	7	39K	230K	Urban (Driving)	RGB & LiDAR
Waymo Open [30]	2020	1K	4	200K	12M	Urban (Driving)	RGB & LiDAR
nuScenes [31]	2020	1K	23	40K	1.4M	Urban (Driving)	RGB & LiDAR
Datasets for 3D Point Cloud Segmentation							
Name and Reference	Year	#Points	#Classes[1]	#Scans	Spatial Size	RGB	Sensors
Oakland [32]	2009	1.6M	5(44)	17	-	N/A	MLS
ISPRS [33]	2012	1.2M	9	-	-	N/A	ALS
Paris-rue-Madame [34]	2014	20M	17	2	-	N/A	MLS
IQmulus [35]	2015	300M	8(22)	10	-	N/A	MLS
ScanNet [11]	2017	-	20(20)	1513	8×4×4	Yes	RGB-D
S3DIS [10]	2017	273M	13(13)	272	10×5×5	Yes	Matterport
Semantic3D [12]	2017	4000M	8(9)	15/15	250×260×80	Yes	TLS
Paris-Lille-3D [36]	2018	143M	9(50)	3	200×280× 30	N/A	MLS
SemanticKITTI [15]	2019	4549M	25(28)	23201/20351	150×100×10	N/A	MLS
Toronto-3D [37]	2020	78.3M	8(9)	4	260×350× 40	Yes	MLS
DALES [38]	2020	505M	8(9)	40	500×500×65	N/A	ALS

Popular ones

- ModelNet40 (classification)
- ShapeNet (classification, segmentation)
- ModelNet10 (subset of ModelNet40, classification)
- Sydney (Classification)
- S3DIS (semantic segmentation)
- ScanNet (segmentation)
- Semantic3D, SemanticKITTI (segmentation)
- KITTI, nuScenes, Waymo (object detection)

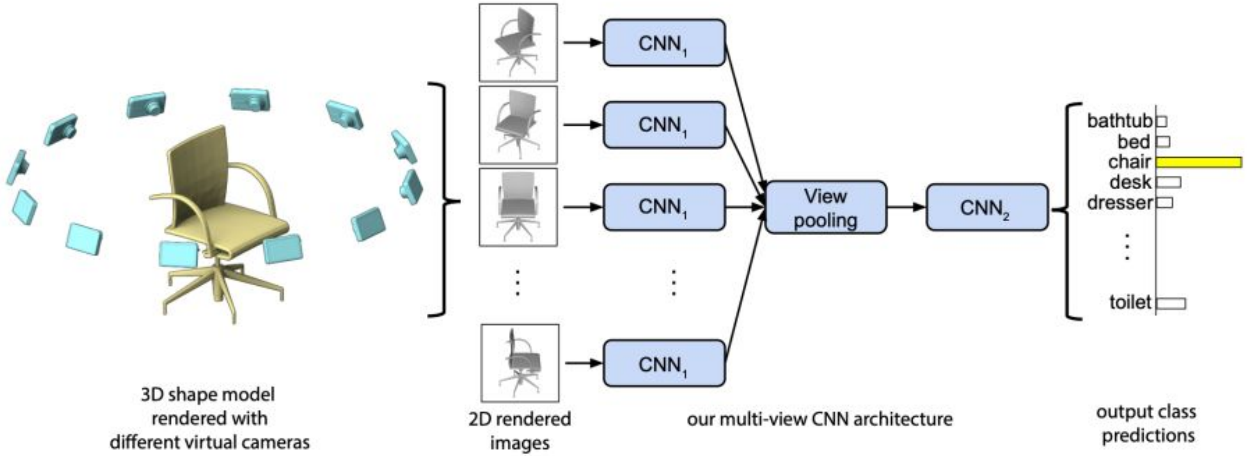
3D data representation

- View (image)-based methods
- Volumetric-based methods
- VoxNet
- Point-based methods
- PointNet
- Graph-based methods
- DGCNN
- Mesh-based methods
- MeshCNN

View-based

Multi-view CNN

3D -> 2D Projections -> Neural Nets (2D CNN)



Multi-view Convolutional Neural Networks for 3D Shape Recognition, ICCV 2015.

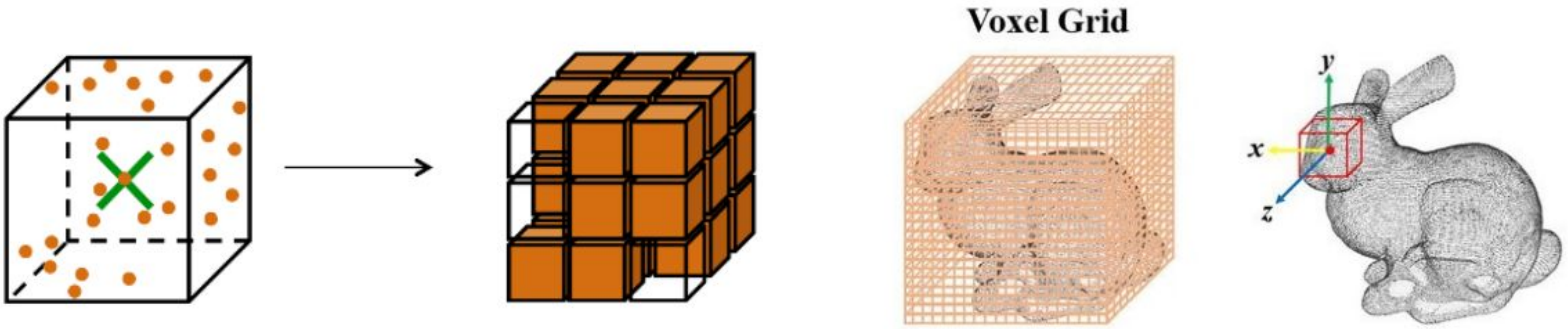
Question

Can we use CNNs without 2D-3D projection?

3D convolution?

Volumetric-based Approaches

Voxelization

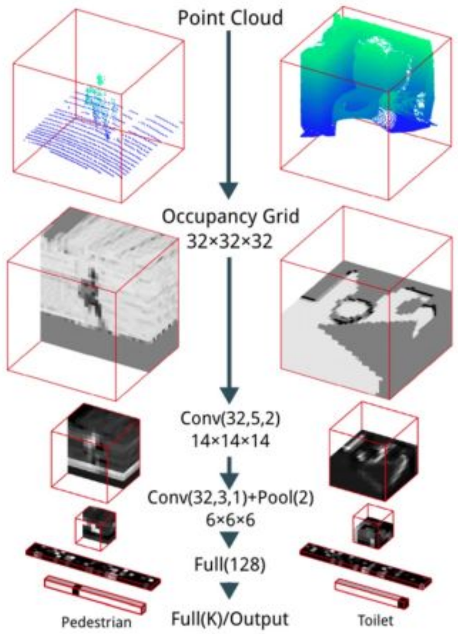


Volumetric-based Approaches

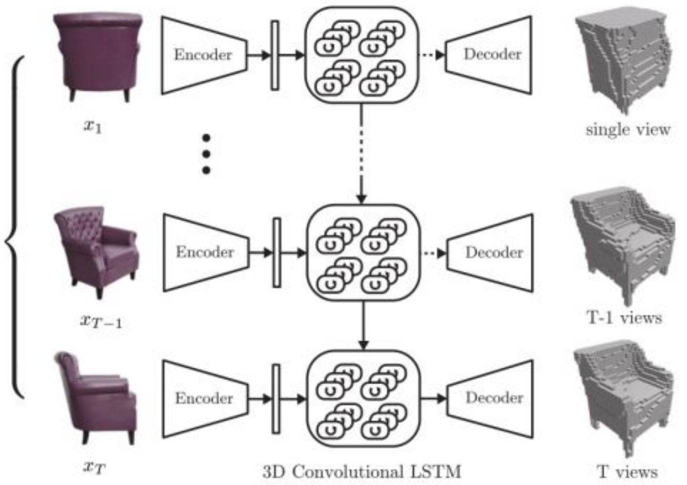
3D Points -> Voxels -> Neural Nets (3D CNN)

• VoxNet (Classification and Segmentation), IROS 2015
Integrating a volumetric Occupancy Grid representation with a supervised 3D Convolutional Neural Network (3D CNN).
• 3D-R2N2 (3D Recurrent Reconstruction Neural Network), ECCV 2016
Given one or multiple views of an object, the network generates voxelized reconstruction of the object in 3D.
• 3DGAN (Generation), NeurIPS 2016
Generates 3D objects from a probabilistic space by leveraging advances in volumetric CNN and GANs.
• VoxelNet (Detection), CVPR 2018
A generic 3D detection network that unifies feature extraction and bounding box prediction into a single stage, end-to-end trainable deep network.

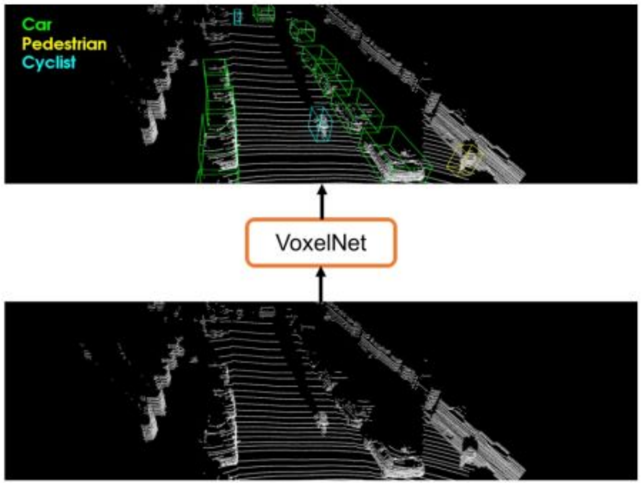
VoxNet



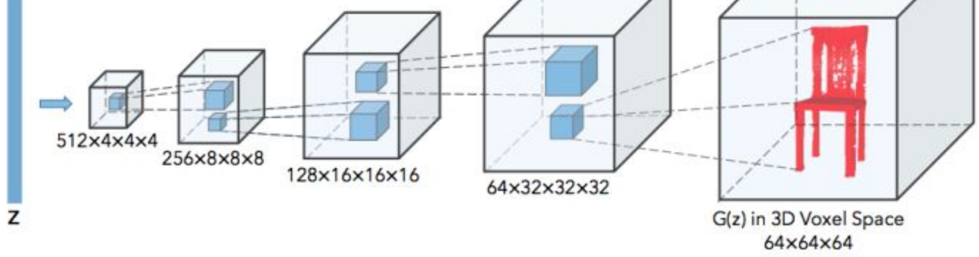
3D-R2N2



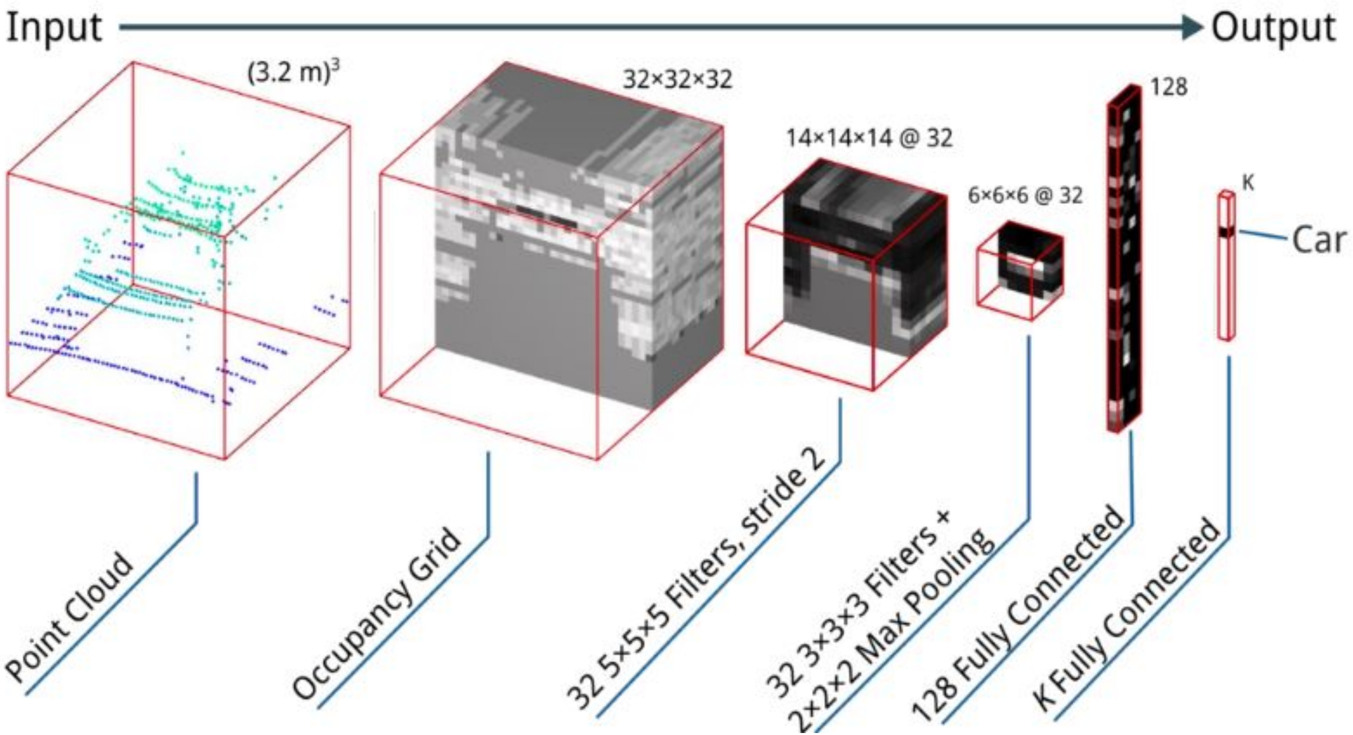
VoxelNet



3DGAN



VoxNet

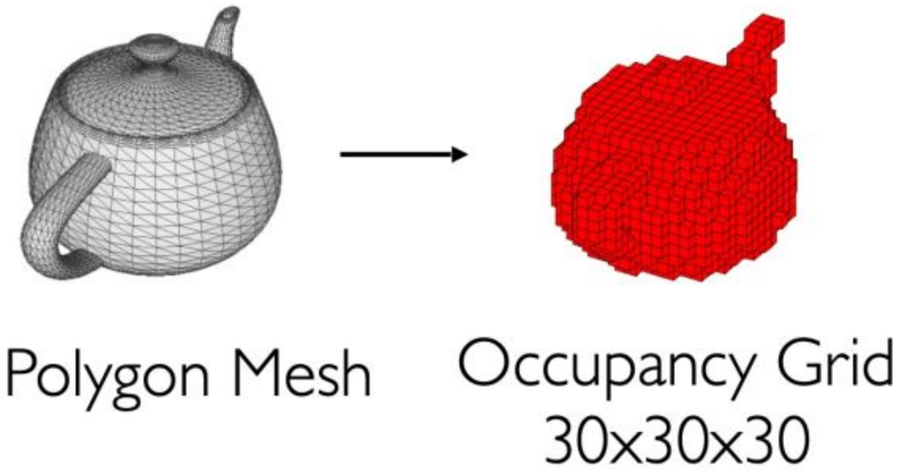


VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition , IROS 2015.

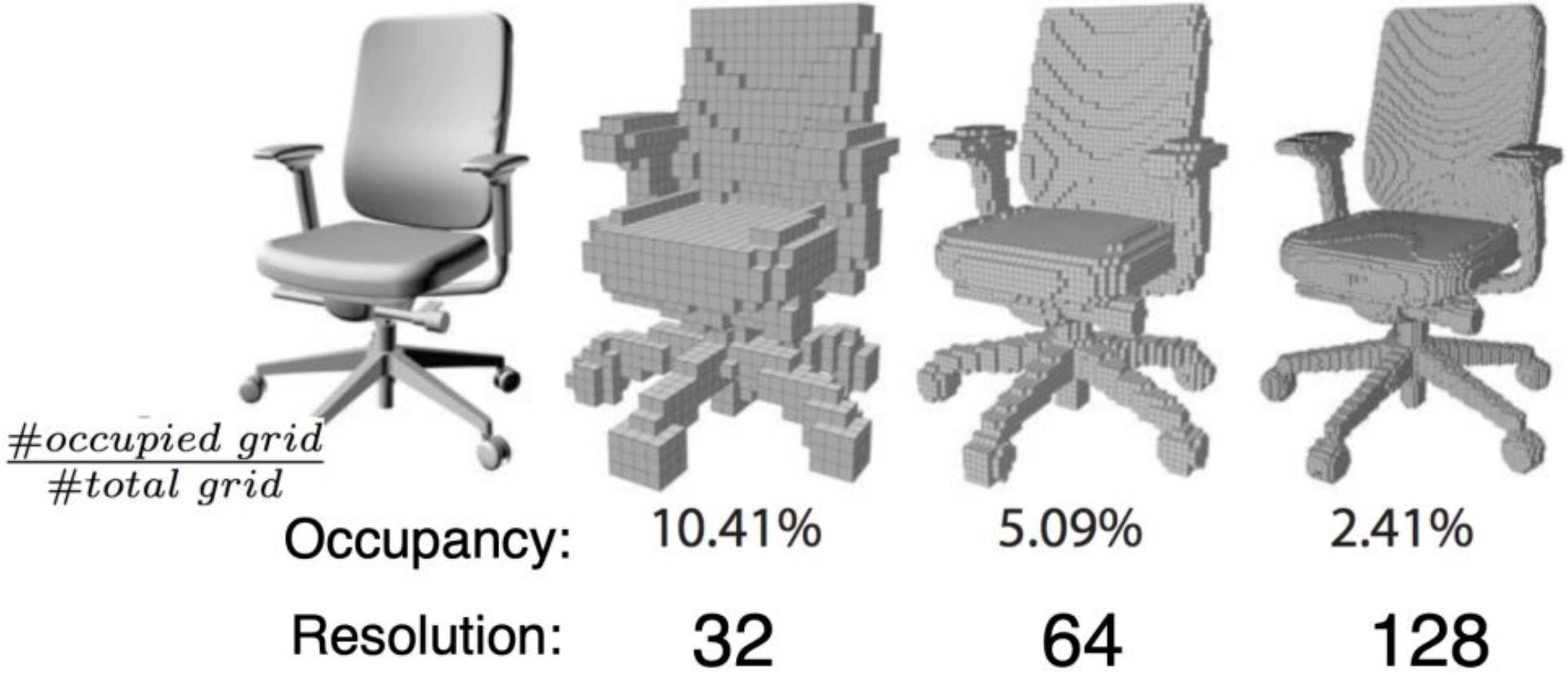
Problems with voxelization

- Memory (1024x1024x1024x1024)
- Lots of zeros
- Resolution
- Quantization artifacts due to continuous to discrete conversion

Quantization Artifacts and Memory Issue



Information loss in voxelization



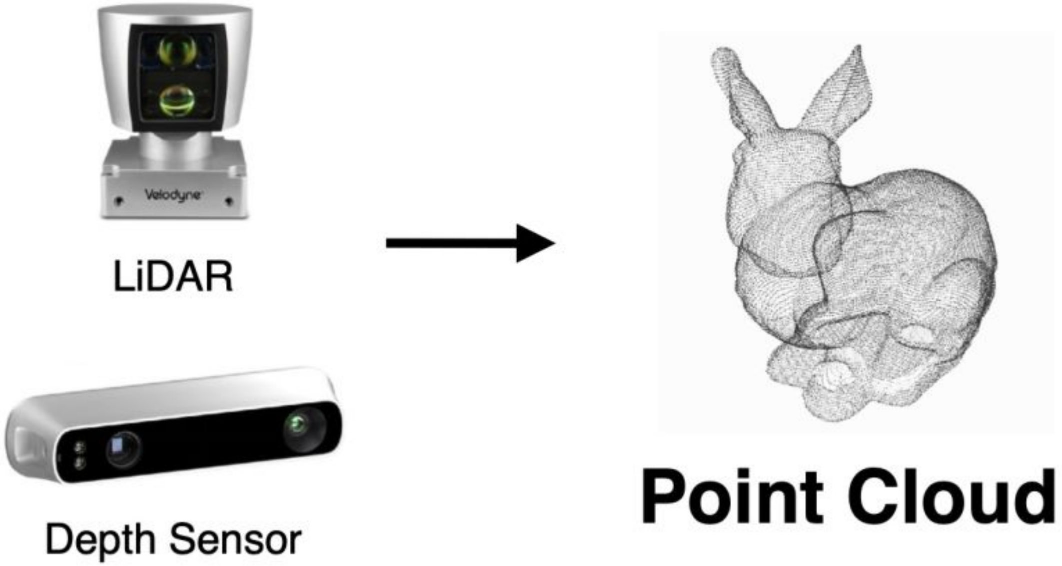
#occupied grid / #total grid
Occupancy:
10.41%
5.09%
2.41%
Resolution:
32
64
128

Research Question

Can we achieve effective feature learning directly on point clouds?

Point Cloud

- The most common 3D sensor data
- Canonical model



Point-based Methods

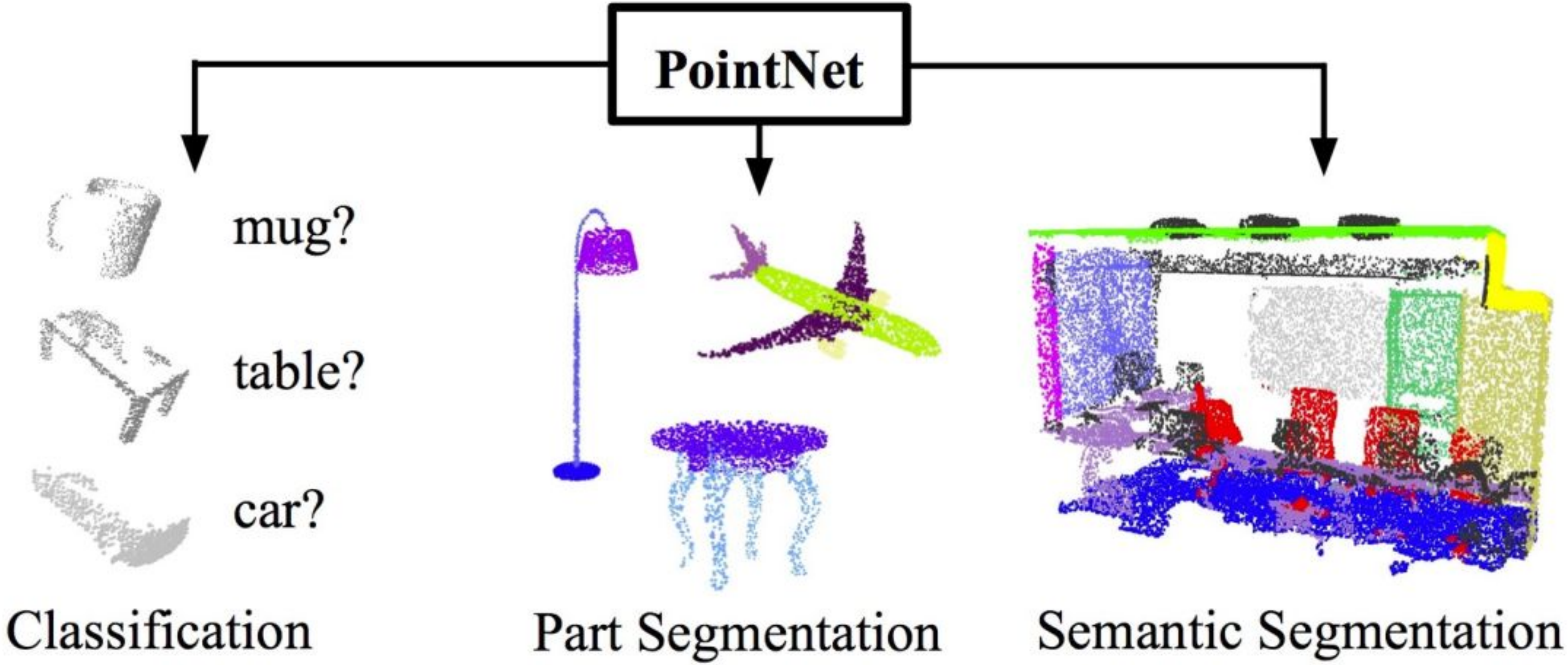
3D Points -> Neural Nets

- **PointNet, CVPR 2017**

  Directly manipulating raw point cloud data and considers global geometry. Achieves permutation invariance of points by operating on each point independently
- **PointNet++, NeurIPS 2017**

  Hierarchical neural network that applies PointNet recursively on a nested partitioning of the input point set. Able to learn local features.

PointNet: Various Tasks



Challenges

- Unordered point set as input

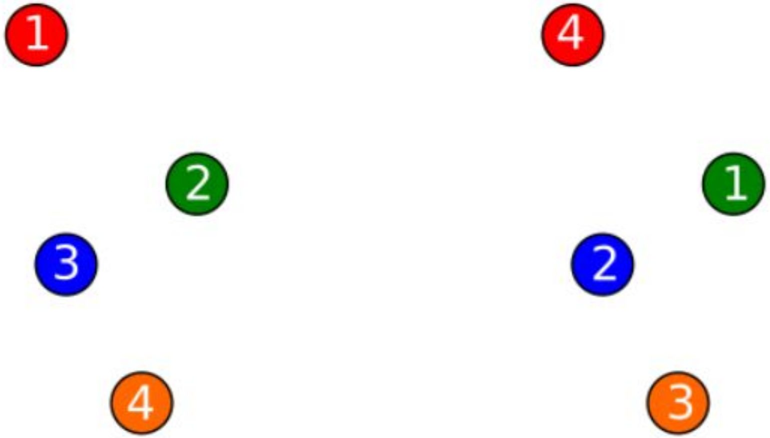
  Model needs to be invariant to N! permutations.

- Invariance under geometric transformations

  Point cloud rotations should not alter classification results.

Unordered Point Set

- Point cloud: N orderless points, each represented by a D dim vector
- Model needs to be invariant to N! permutations



Permutation invariance: Symmetric Function

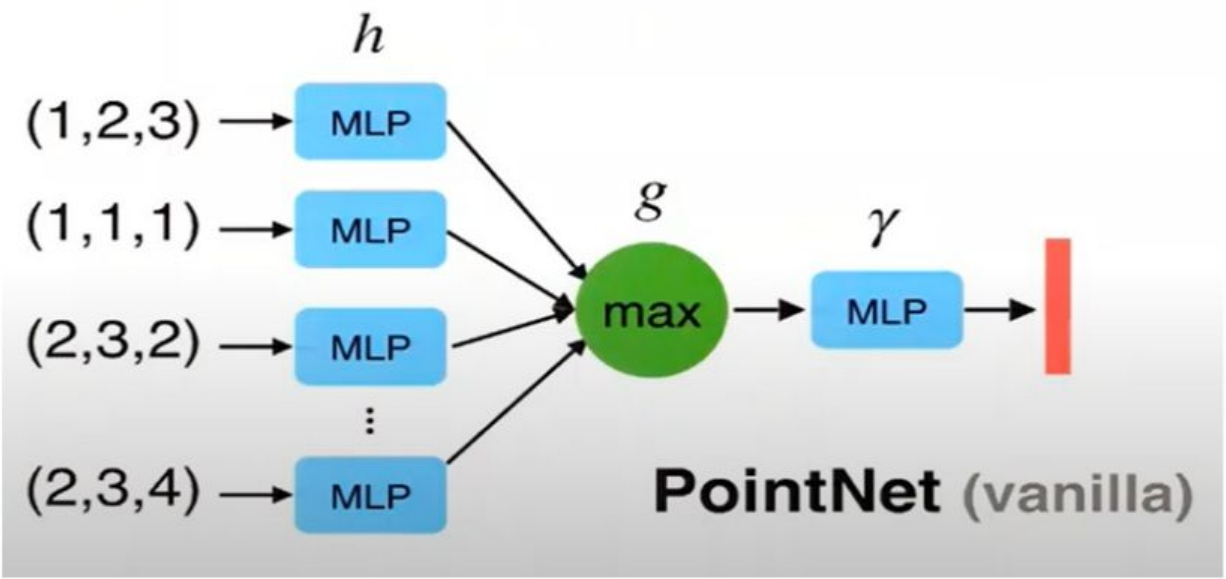
- How to construct a family of symmetric functions by neural networks?
- Examples:

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

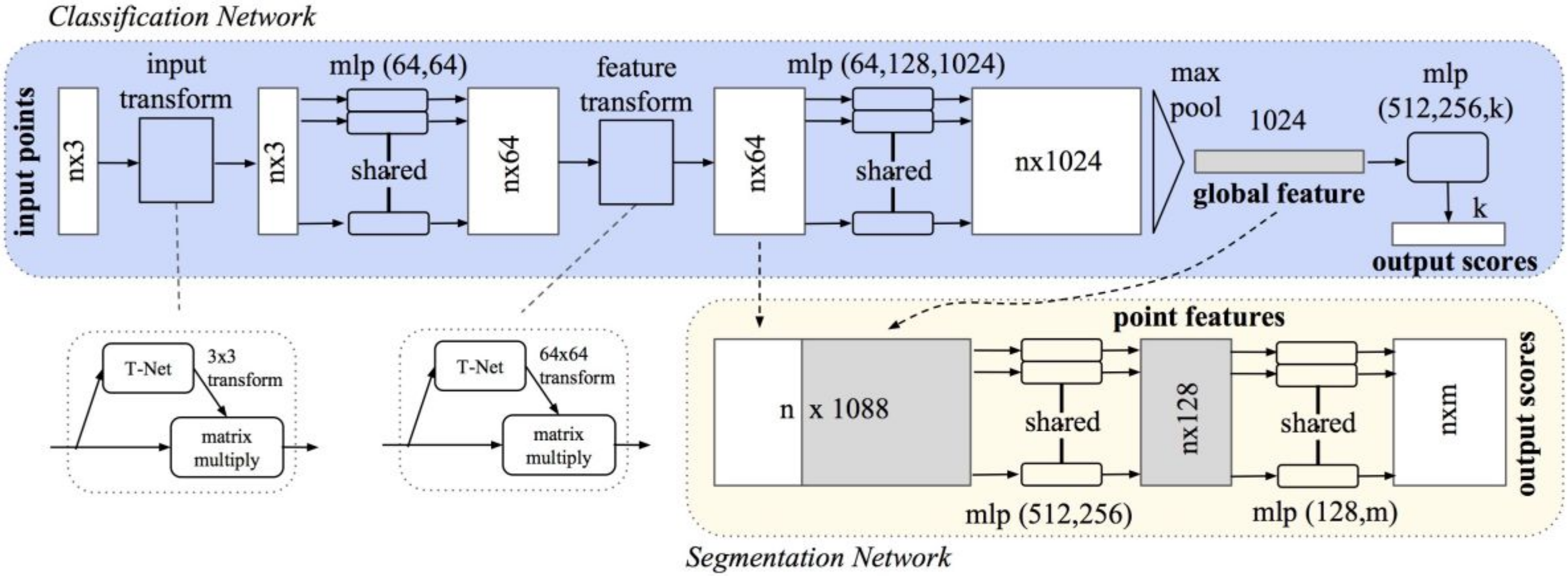
$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

Basic PointNet Architecture

- PointNet uses multi-layer perceptron(MLP) and max pooling:



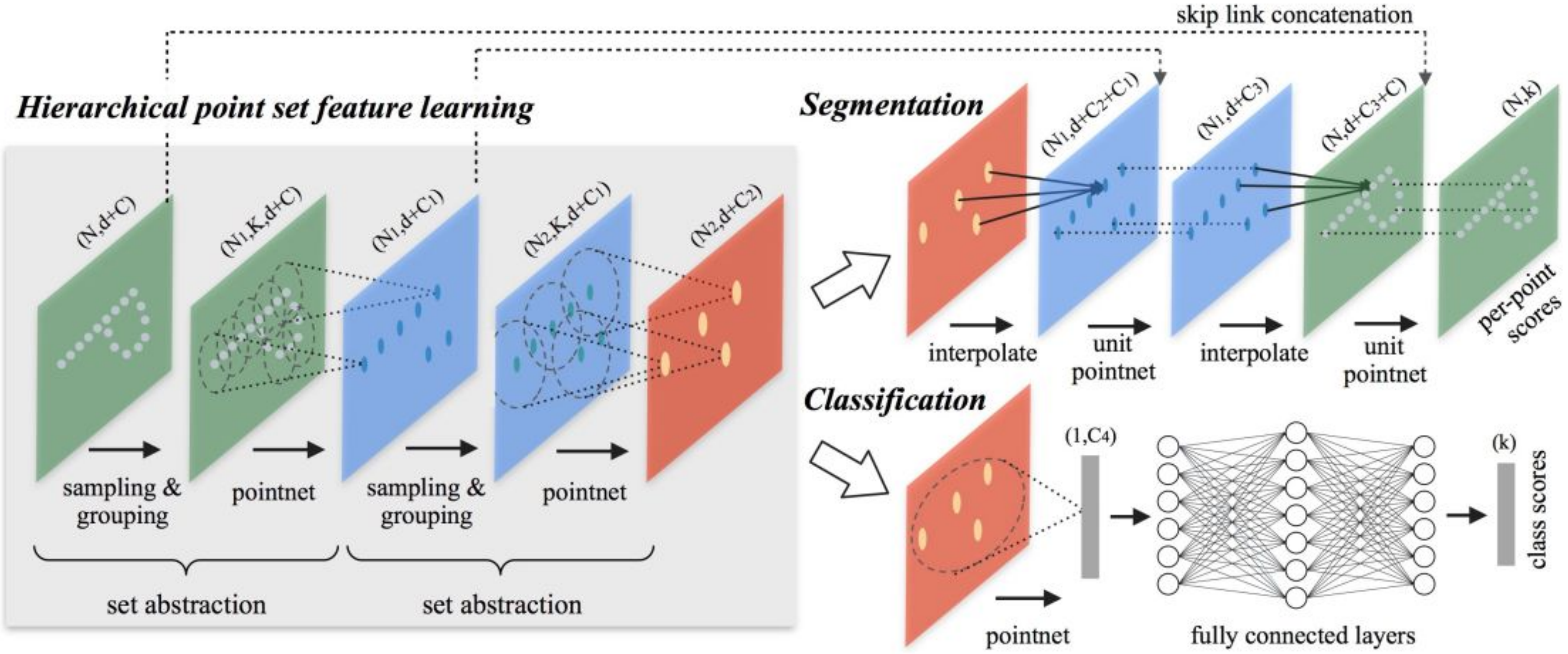
PointNet: Classification & Segmentation Network



Limitations of PointNet

- No local context for each point!
  - Cannot capture local information
- Works on individual points
- Ignores the geometric points
- Global feature depends on absolute coordinate. Hard to generalize to unseen scene configurations.

PointNet++



Qi, Charles R., et al. **"PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space."**, NeurIPS 2017

Resources: Github: awesome-point-cloud-analysis

README.md

awesome-point-cloud-analysis awesome

for anyone who wants to do research about 3D point cloud.
I will try to update this list everyday!!!

```
- Recent papers (from 2017)
```

Table of Contents

Keywords

`dat.` : dataset | `cls.` : classification | `rel.` : retrieval | `seg.` : segmentation
`det.` : detection | `tra.` : tracking | `pos.` : pose | `dep.` : depth
`reg.` : registration | `rec.` : reconstruction | `aut.` : autonomous driving
`oth.` : other, including normal-related, correspondence, mapping, matching, alignment, compression, generative model...

Statistics: 🔥 code is available & stars >= 100 | ⭐ citation >= 50

Graph-based Approaches for Point Clouds

- Points -> Nodes
- Neighbourhoods -> Edges
- Point Convolution as Graph Convolution
- Graph CNN for point cloud processing
- DGCNN
    - Graph-based Point Cloud Learning Architecture
    - Computes graph on point cloud
    - Constructs graph dynamically
    - The local structure itself is informative

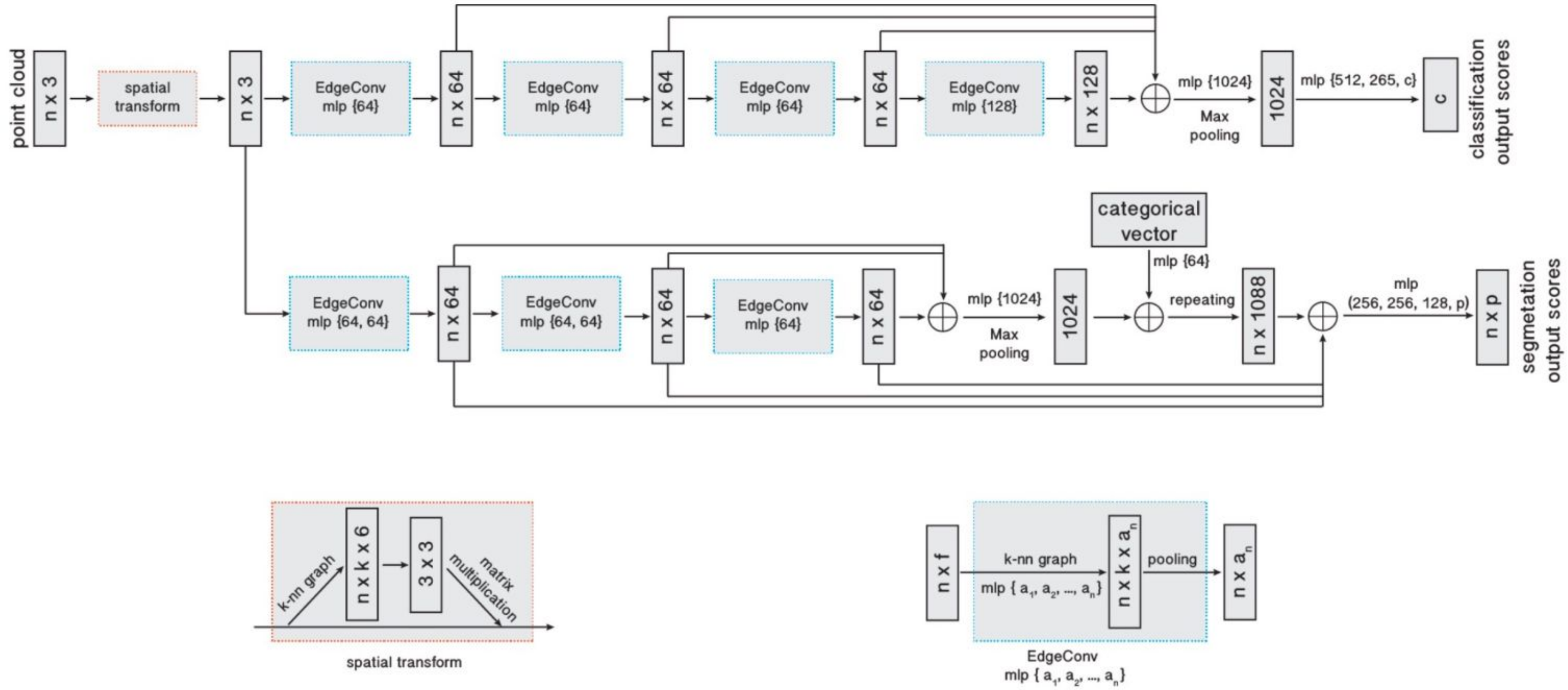


Fig. 3. **Model architectures:** The model architectures used for classification (top branch) and segmentation (bottom branch). The classification model takes as input n points, calculates an edge feature set of size k for each point at an EdgeConv layer, and aggregates features within each set to compute EdgeConv responses for corresponding points. The output features of the last EdgeConv layer are aggregated globally to form an $1D$ global descriptor, which is used to generate classification scores for c classes. The segmentation model extends the classification model by concatenating the $1D$ global descriptor and all the EdgeConv outputs (serving as local descriptors) for each point. It outputs per-point classification scores for p semantic labels. $\oplus$: concatenation. **Point cloud transform block:** The point cloud transform block is designed to align an input point set to a canonical space by applying an estimated 3×3 matrix. To estimate the 3×3 matrix, a tensor concatenating the coordinates of each point and the coordinate differences between its k neighboring points is used. **EdgeConv block:** The EdgeConv block takes as input a tensor of shape $n \times f$, computes edge features for each point by applying a multi-layer perceptron (mlp) with the number of layer neurons defined as $\{a_1, a_2, \ldots, a_n\}$, and generates a tensor of shape $n \times a_n$ after pooling among neighboring edge features.

EdgeConv Operation

- Concatenate the point features, pass the concatenated feature through a MLP to get edge features
- A symmetric aggregation function is applied upon the edge features to get new point feature

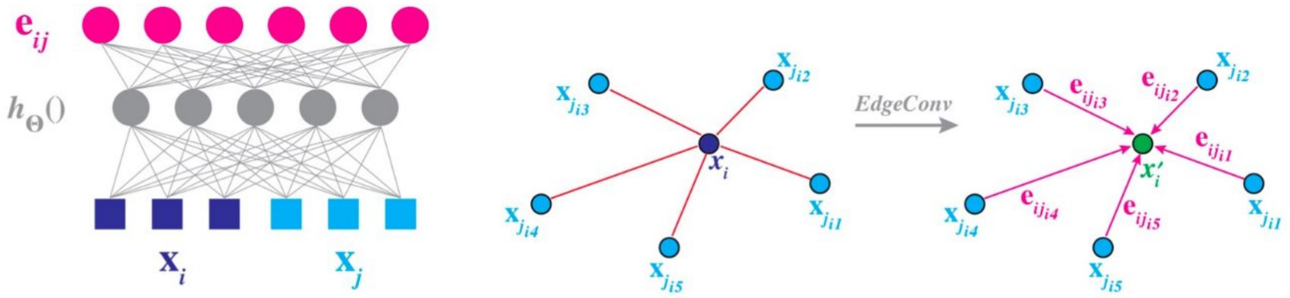
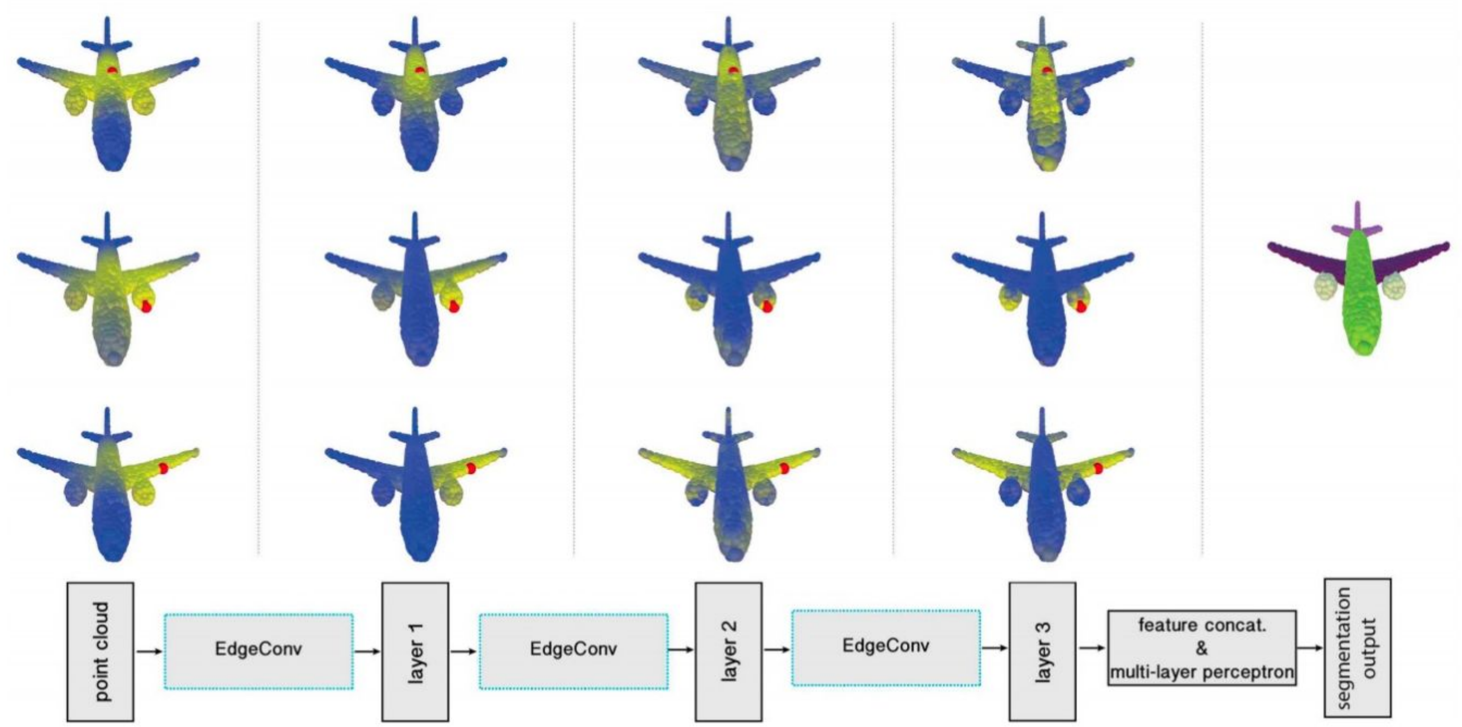
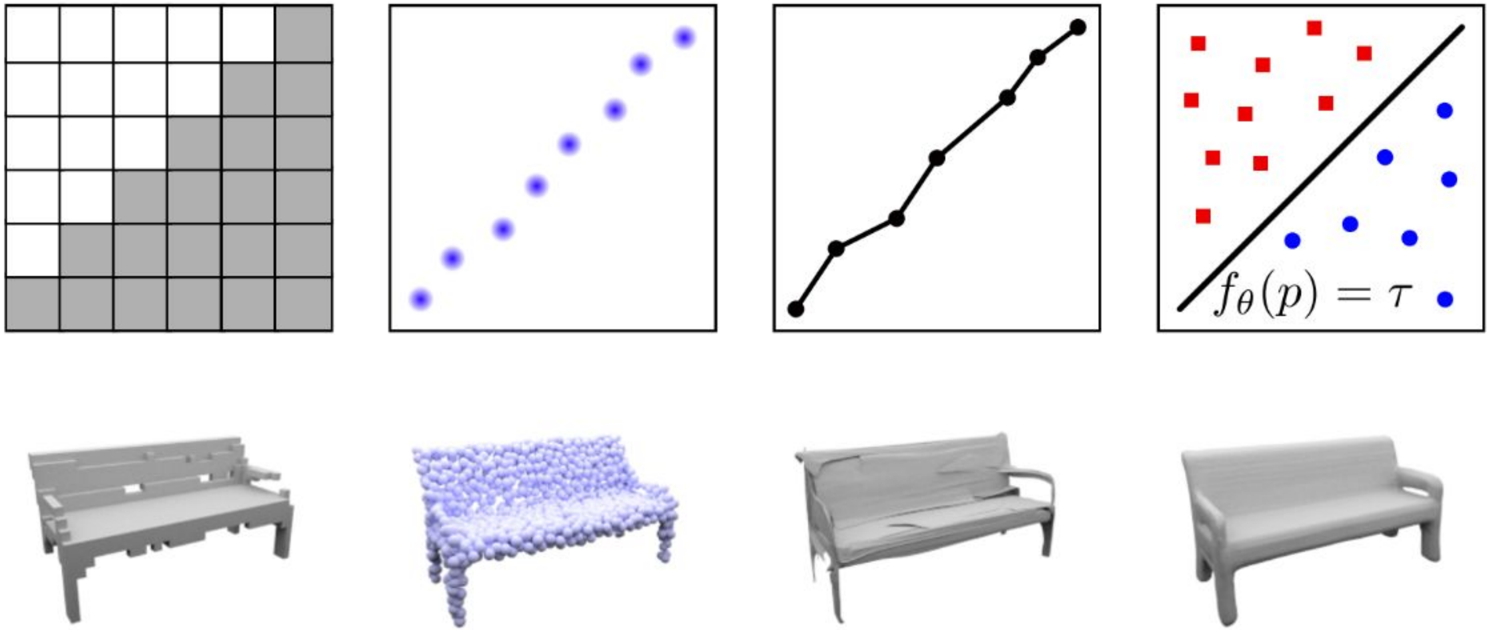


Fig. 2. **Left**: Computing an edge feature, $\mathbf{e}_{ij}$ (top), from a point pair, $\mathbf{x}_i$ and $\mathbf{x}_j$ (bottom). In this example, $h_\Theta()$ is instantiated using a fully connected layer, and the learnable parameters are its associated weights. **Right**: The EdgeConv operation. The output of EdgeConv is calculated by aggregating the edge features associated with all the edges emanating from each connected vertex.

DGCCN: Learning in Layers



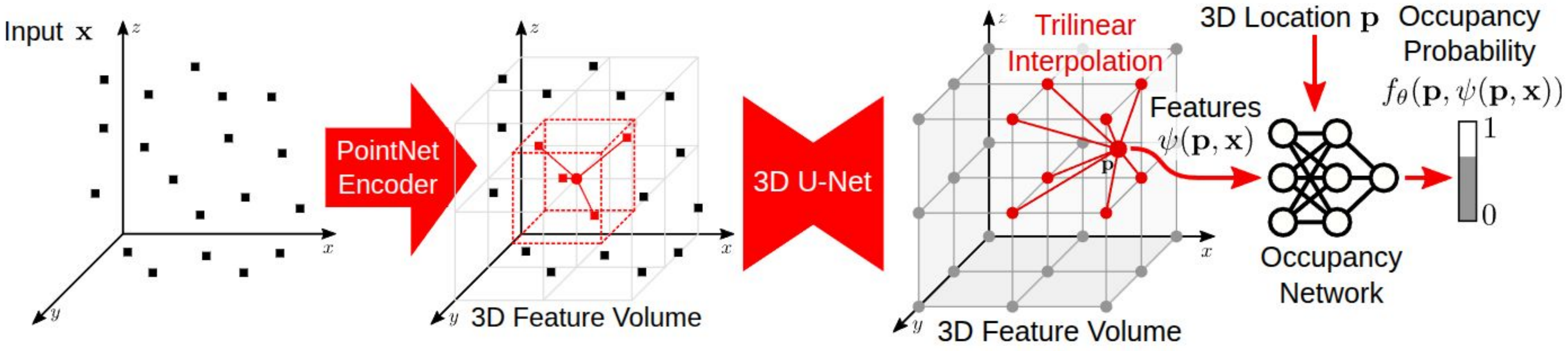
3D Representations



- Traditional Explicit Representations $\Rightarrow$ **Discrete**
- Implicit Neural Representation $\Rightarrow$ **Continuous**

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

Convolutional Occupancy Networks



- **3D Volume Encoder:** Local PointNet processes input, volumetric feature encoding
- **3D Volume Decoder:** Processed by 3D U-Net, query features via trilinear interp.
- **Occupancy Readout:** Shallow occupancy network $f_\theta(\cdot)$

Peng, Niemeyer, Mescheder, Pollefeys and Geiger: Convolutional Occupancy Networks. arXiv, 2020.

Level Sets : https://mathinsight.org/level_sets

Resources used for these slides and additional resources will be added in the github page.

Thank you!!