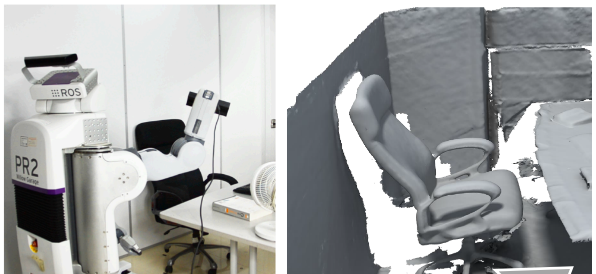
CSE 152: Computer Vision

Hao Su

Lecture 12: 3D Deep Learning

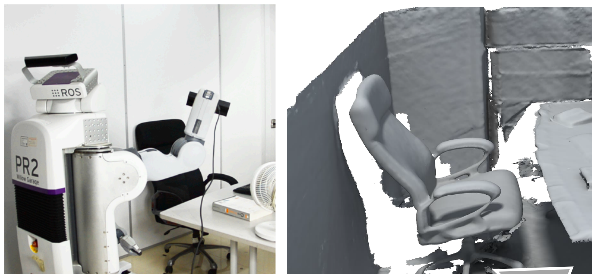
Credit: Stanford CS231n, L13

Broad Applications of 3D data

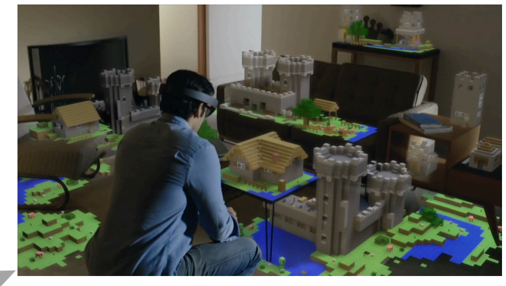


Robotics

Broad Applications of 3D data

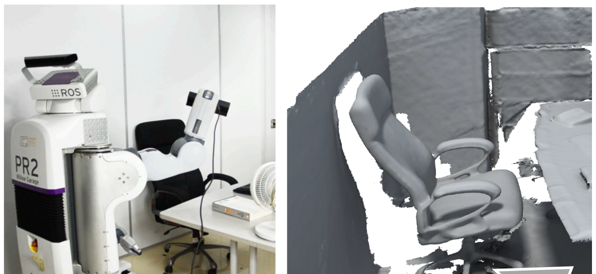


Robotics

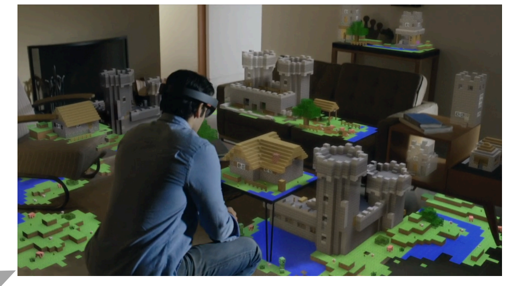


Augmented Reality

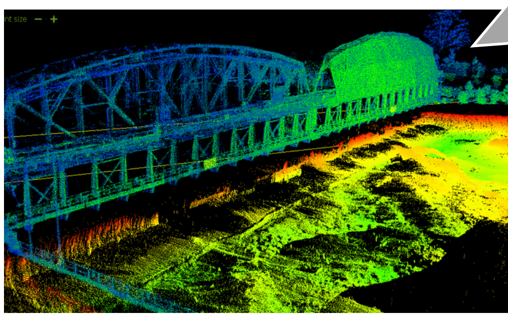
Broad Applications of 3D data



Robotics

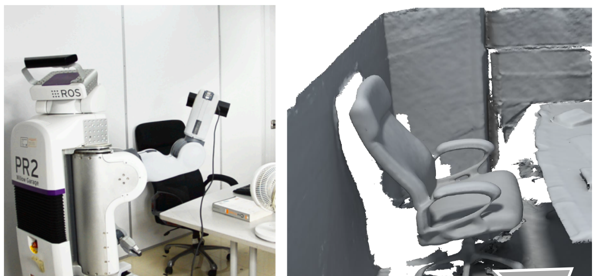


Augmented Reality

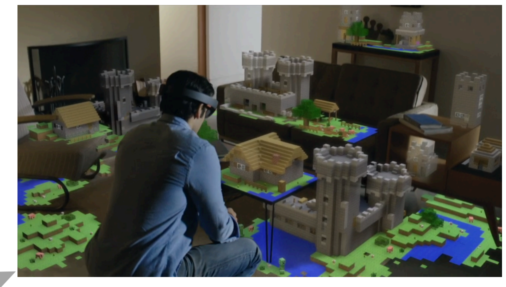


Autonomous driving

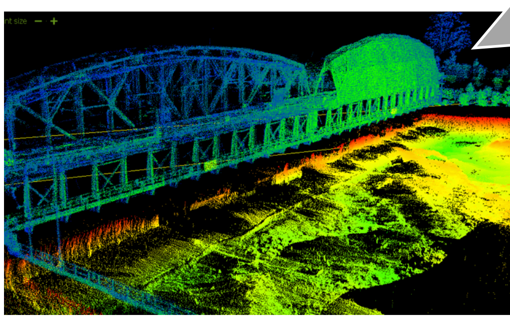
Broad Applications of 3D data



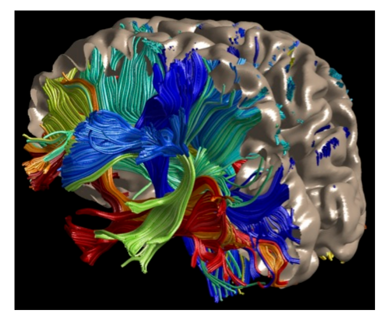
Robotics



Augmented Reality



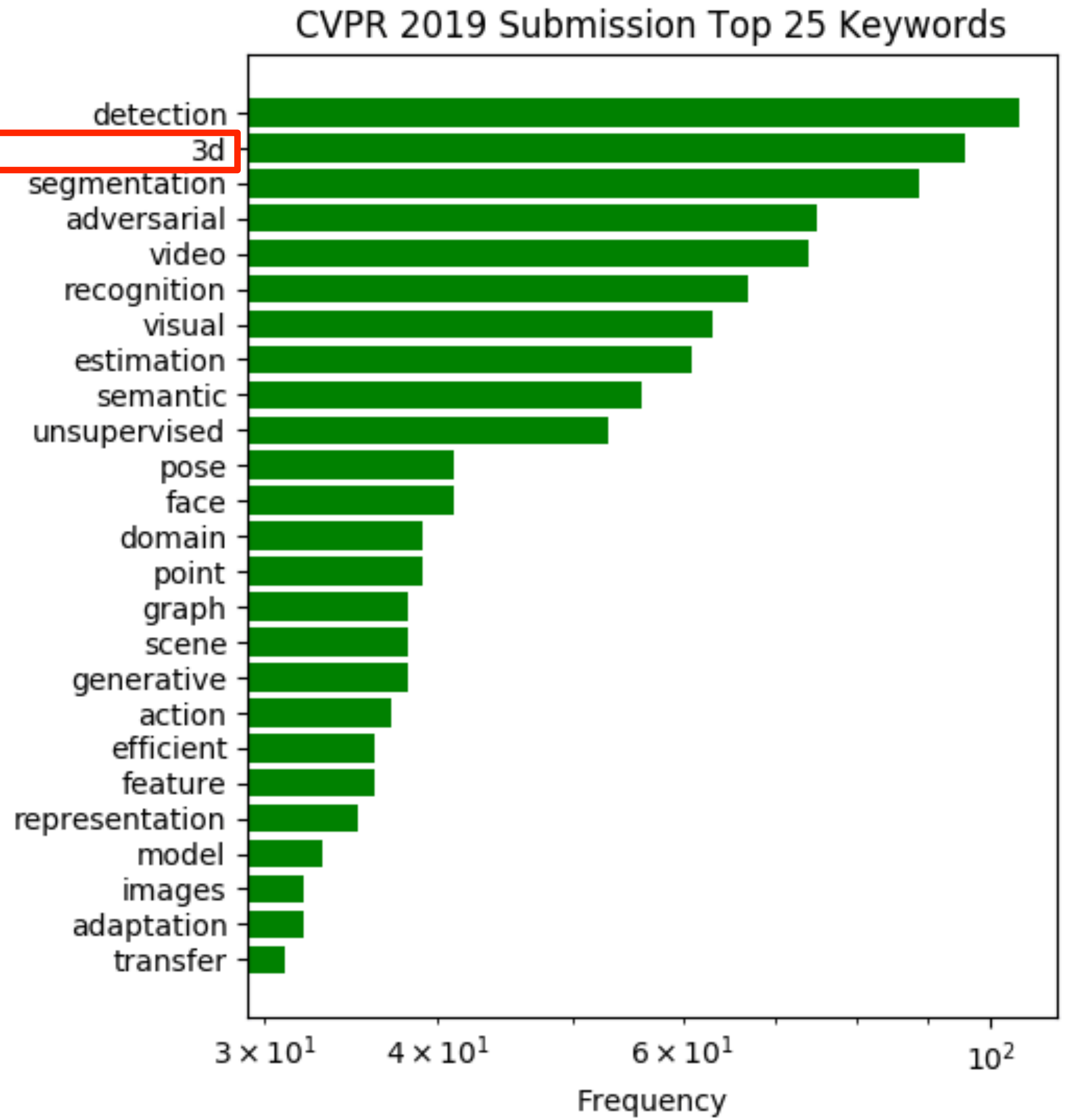
Autonomous driving



Medical Image Processing

Acquire Knowledge of 3D World by Learning

A priori knowledge of
the 3D world

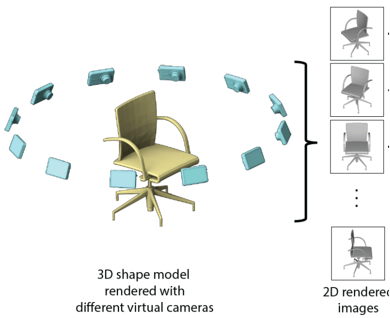

CVPR 2019 Submission Top 25 Keywords
detection
3d
segmentation
adversarial
video
recognition
visual
estimation
semantic
unsupervised
pose
face
domain
point
graph
scene
generative
action
efficient
feature
representation
model
images
adaptation
transfer
3×10^1
4×10^1
6×10^1
10^2
Frequency

The Representation Challenge of 3D Deep Learning

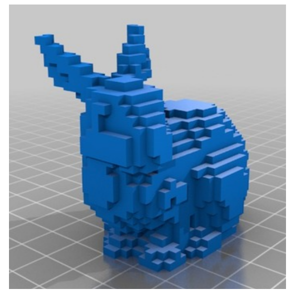
Rasterized form (regular grids)

Geometric form (irregular)

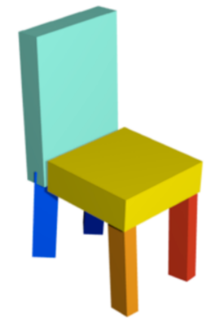
The Representation Challenge of 3D Deep Learning



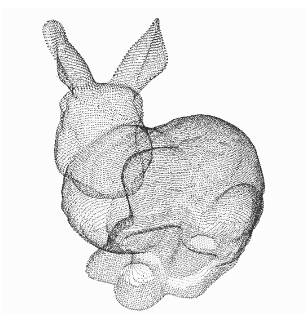
Multi-view



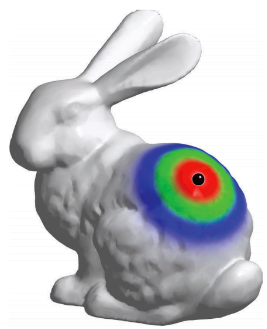
Volumetric



Part Assembly



Point Cloud



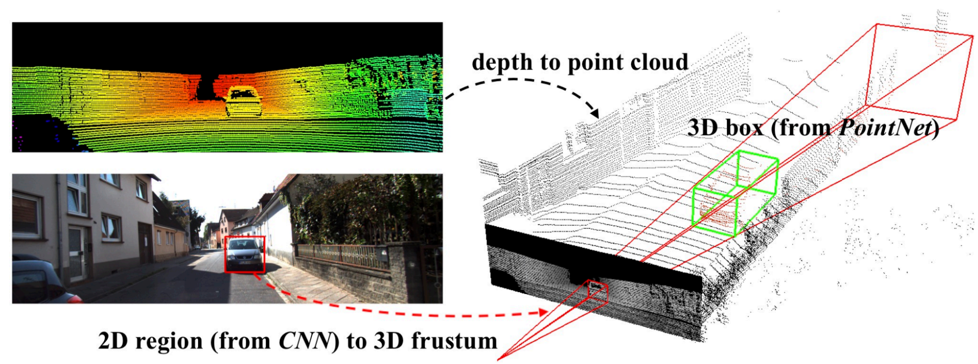
Mesh (Graph CNN)

$$F(x) = 0$$

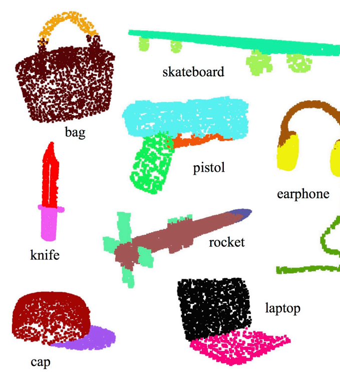
Implicit Shape

The Richness of 3D Learning Tasks

3D Analysis



Detection



Classification

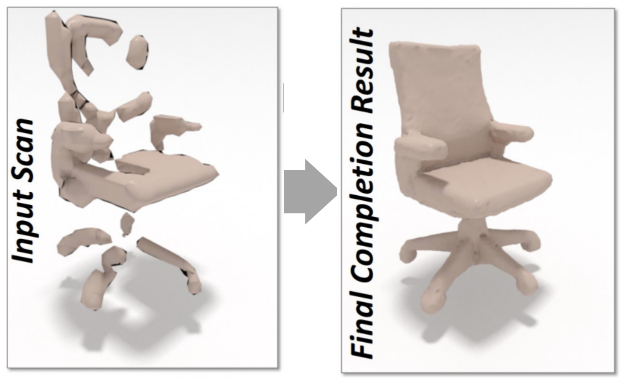
Segmentation (object/scene)

Correspondence

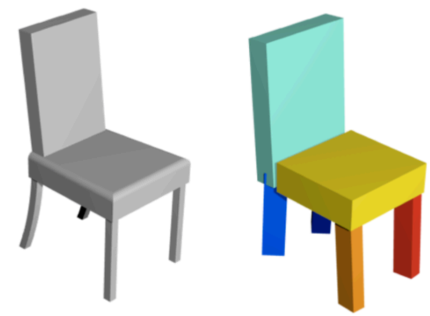
The Richness of 3D Learning Tasks

3D Synthesis

Monocular
3D reconstruction



Shape completion



Shape modeling

Agenda

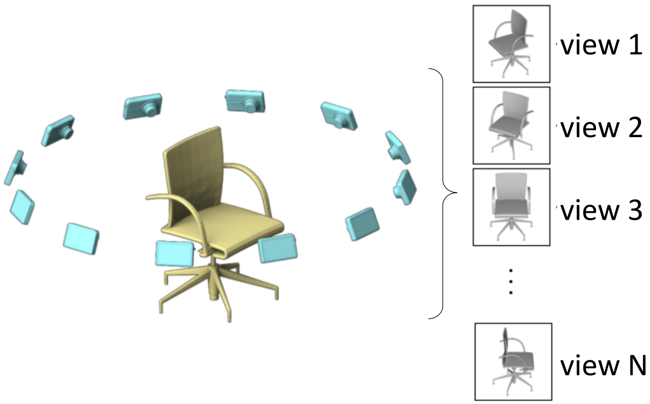
- **3D Classification**
- 3D Reconstruction

Multi-View CNN

Given an Input Shape

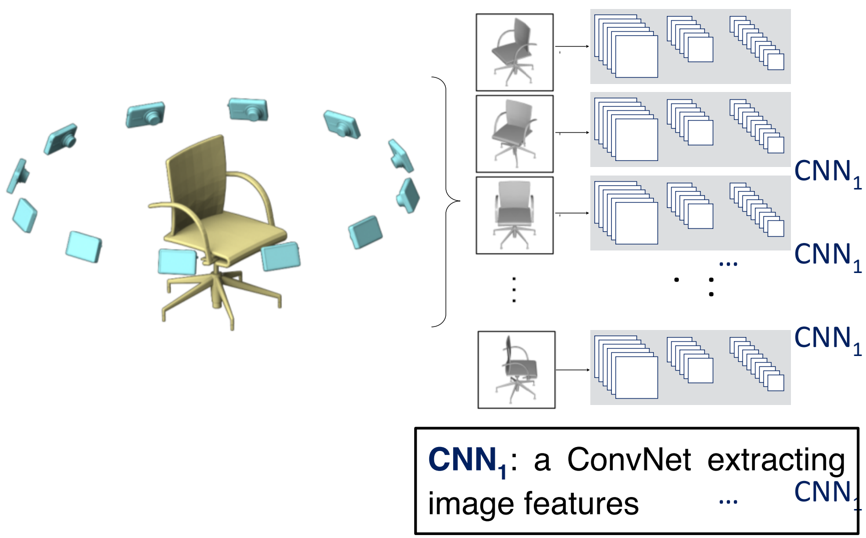
Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

Render with Multiple Virtual Cameras



Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

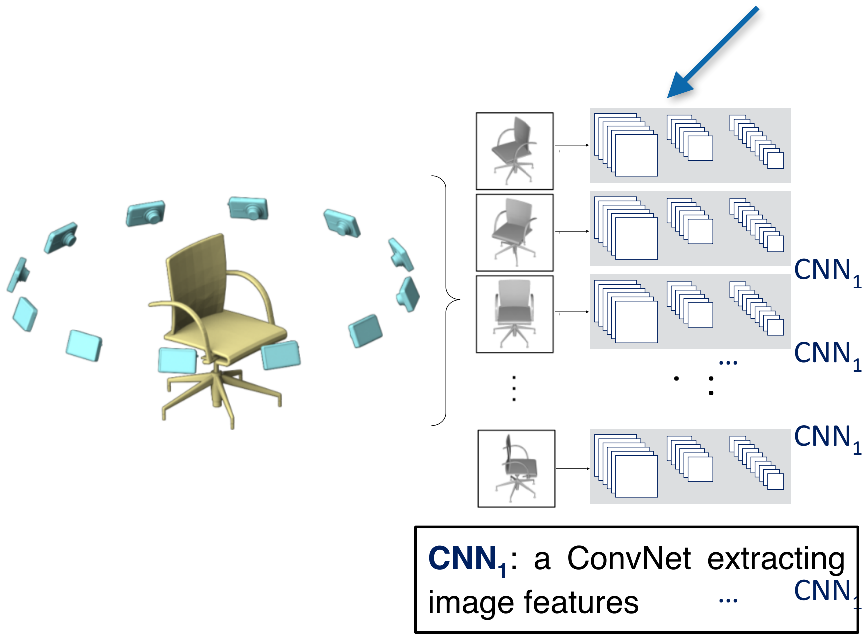
The Rendered Images are Passed through CNN_1 for Image Features



Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

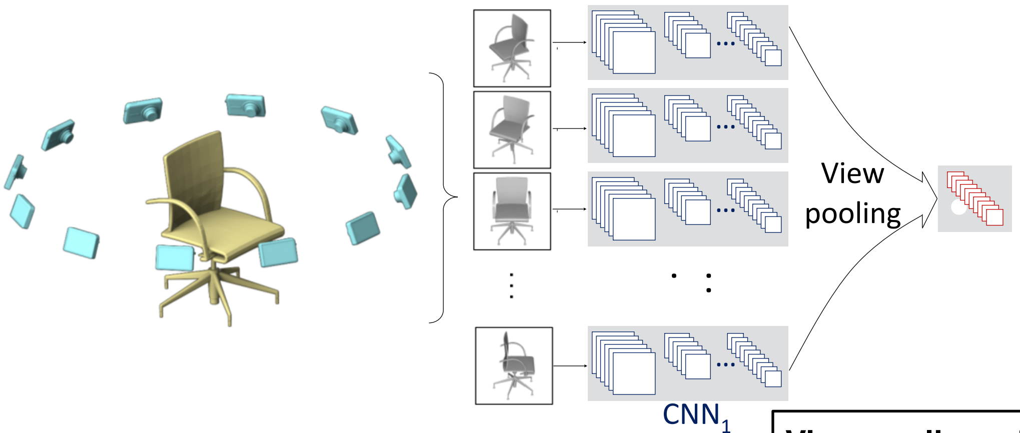
Fine-tuning Pretrained Network Weights

Can reuse “pertrained” weights from image classification networks



Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**”, *ICCV 2015*

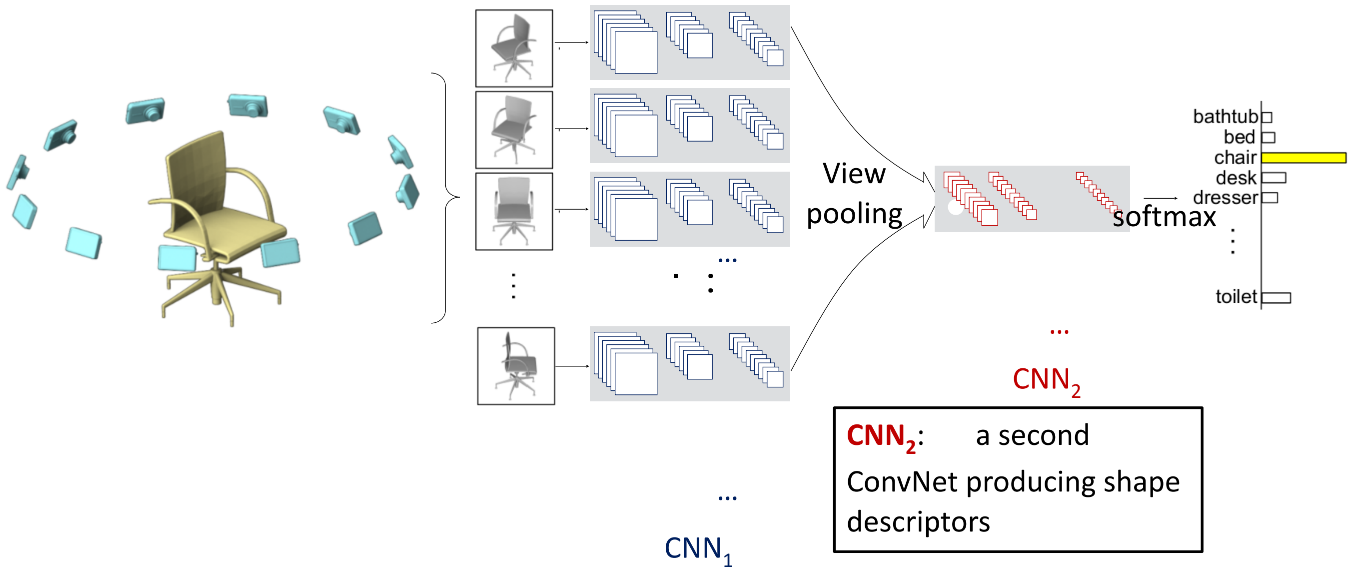
View Pooling



View pooling: element-wise max-pooling across all views

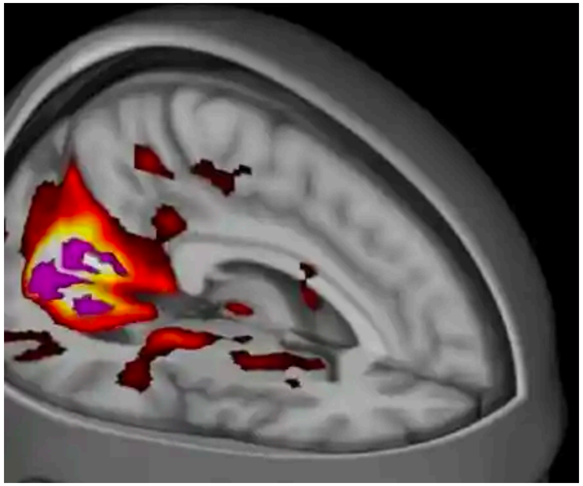
Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

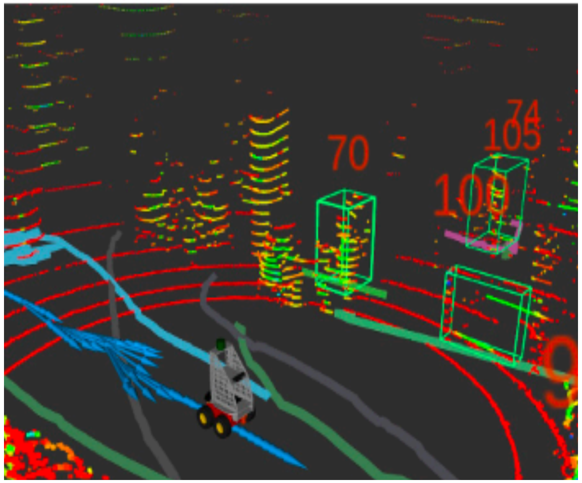
… and then Passed through CNN_2 and to Generate Final Predictions



Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

- Can leverage vast literature of image classification
- Can use pertrained features
- In many scenarios, such as fMRI images and LiDAR data, we are not quite able to “render” 3D into images (self-occlusion)





Volumetric CNN

Can we use CNNs but avoid projecting the 3D data to views first?

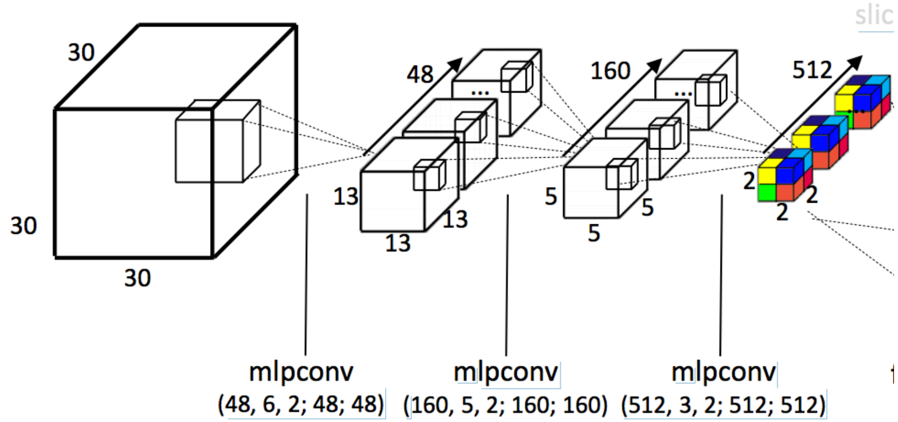
Straight-forward idea: Extend 2D grids 3D grids

Voxelization

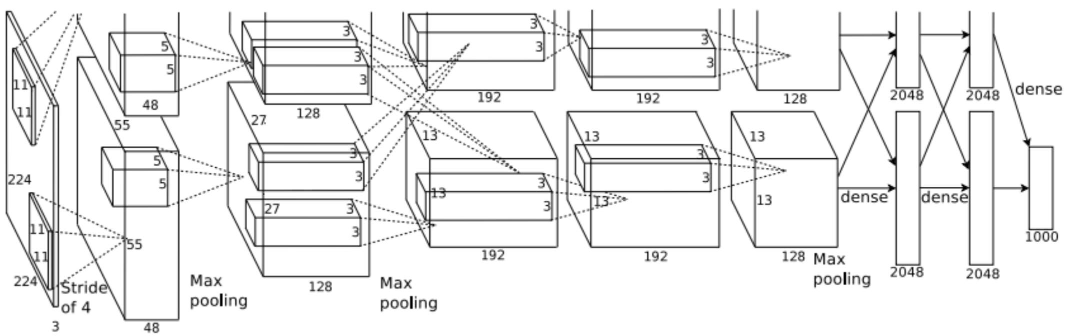
Represent the occupancy of regular 3D grids

3D CNN on Volumetric Data

3D convolution uses 4D kernels



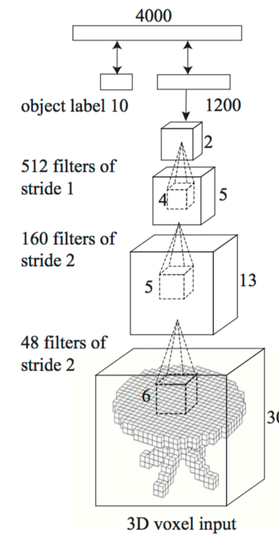
Complexity Issue



AlexNet, 2012

Input resolution: 224x224

224x224=50176

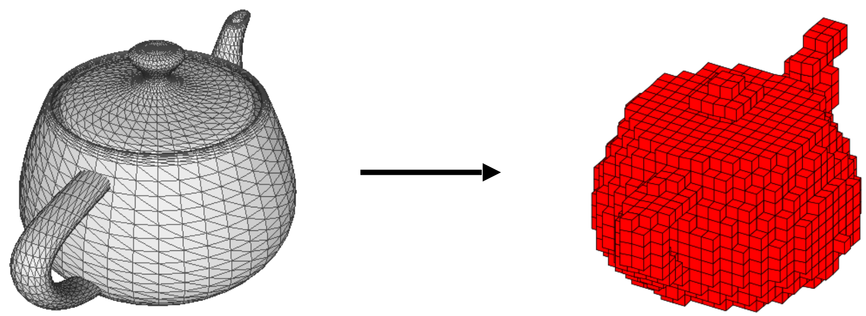


3DShapeNets, 2015

Input resolution: 30x30x30

224x224=27000

Complexity Issue



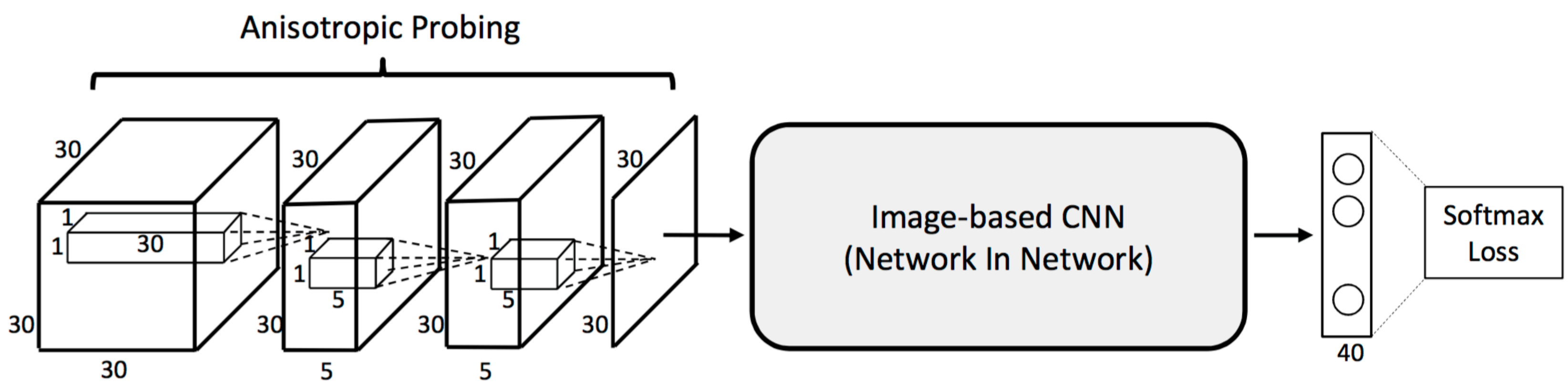
Polygon Mesh

Occupancy Grid
30x30x30

Information loss in voxelization

Idea 1: Learn to Project

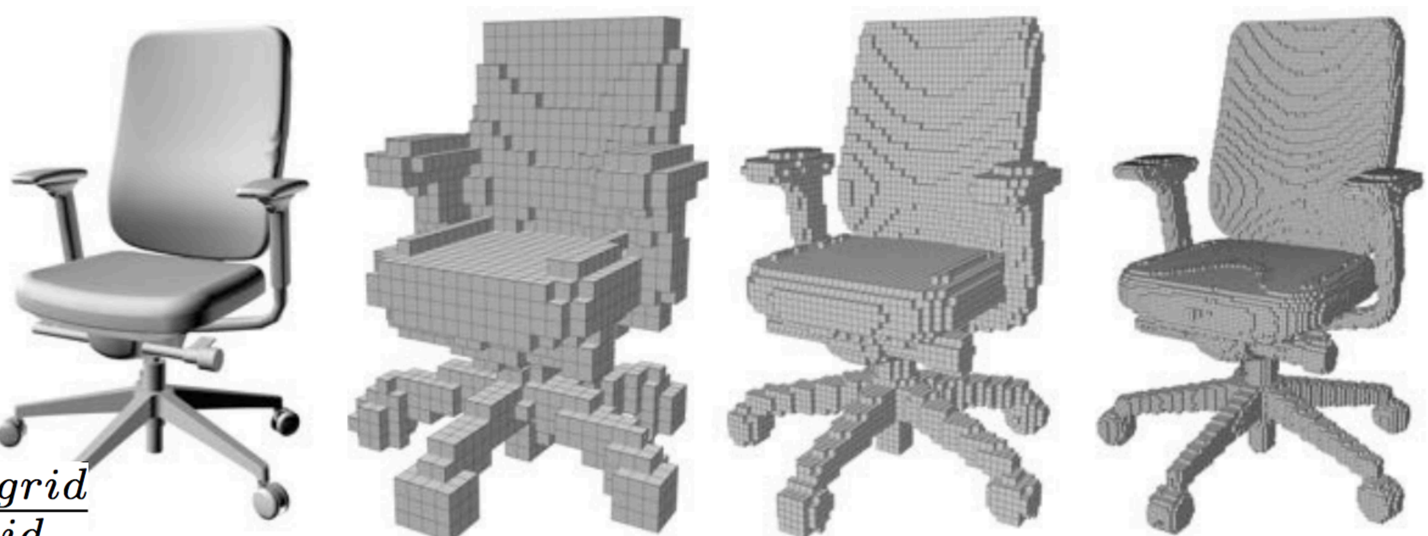
Idea: "X-ray" rendering + Image (2D) CNNs
very low #param, very low computation



Su et al., "**Volumetric and Multi-View CNNs for Object Classification on 3D Data**", *CVPR 2016*

Many other works in autonomous driving that uses **bird's eye view** for object detection

More Principled: Sparsity of 3D Shapes

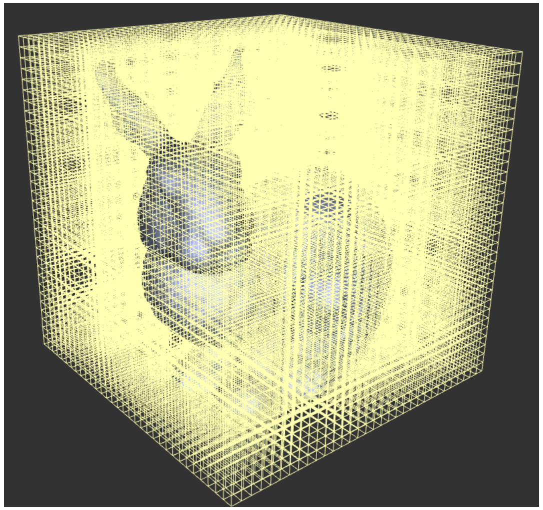


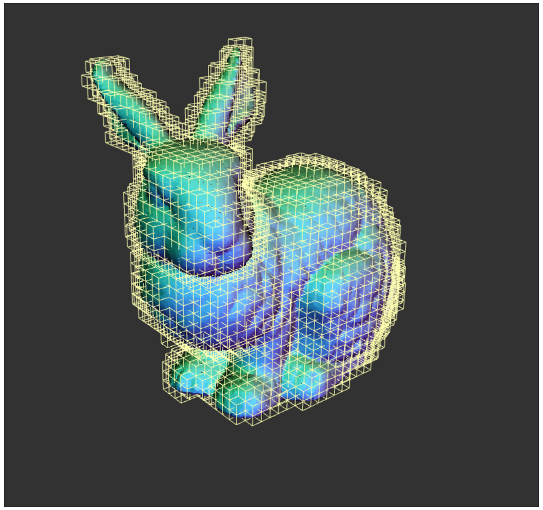
$$\frac{\#occupied\ grid}{\#total\ grid}$$

Occupancy:	10.41%	5.09%	2.41%
Resolution:	32	64	128

Store only the Occupied Grids

- Store the sparse surface signals
- Constrain the computation near the surface

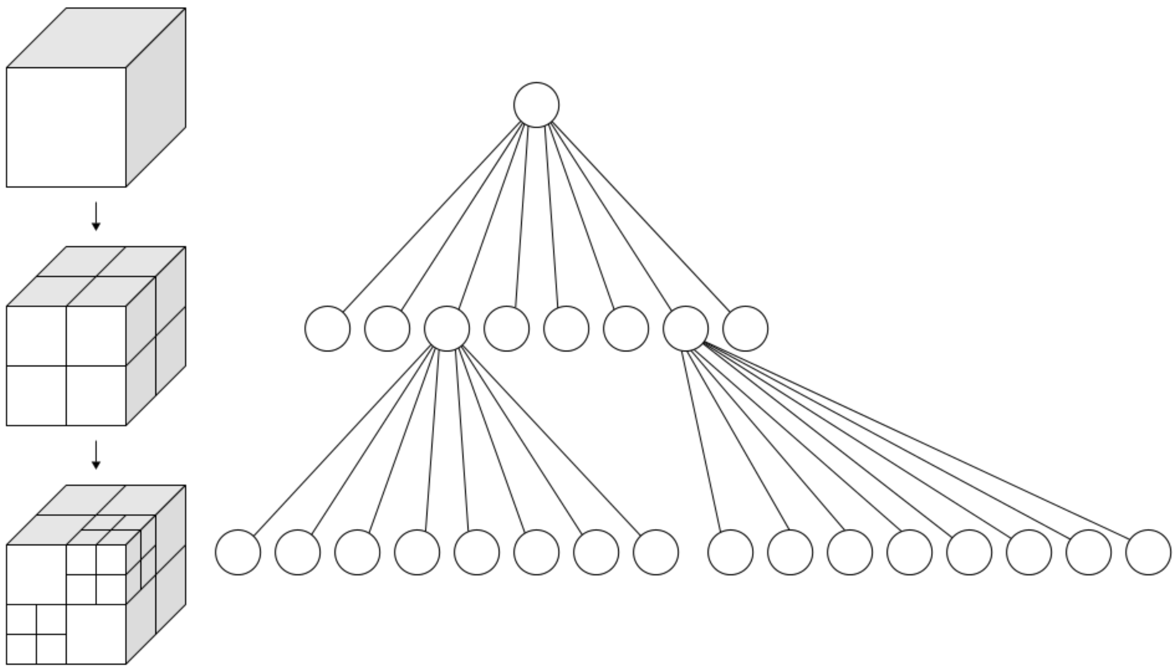




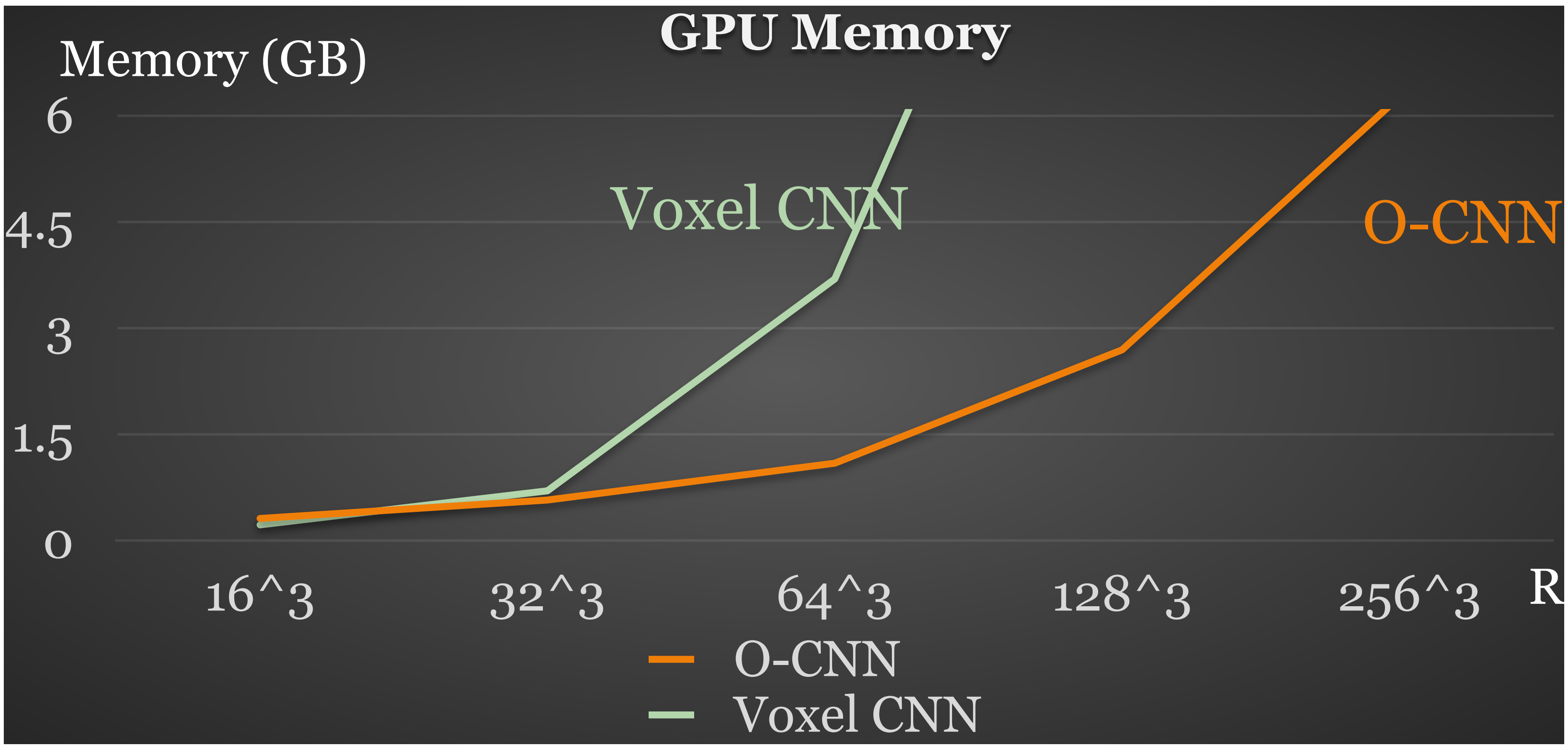
Octree: Recursively Partition the Space

Each internal node has exactly eight children

Neighborhood searching: Hash table



Memory Efficiency

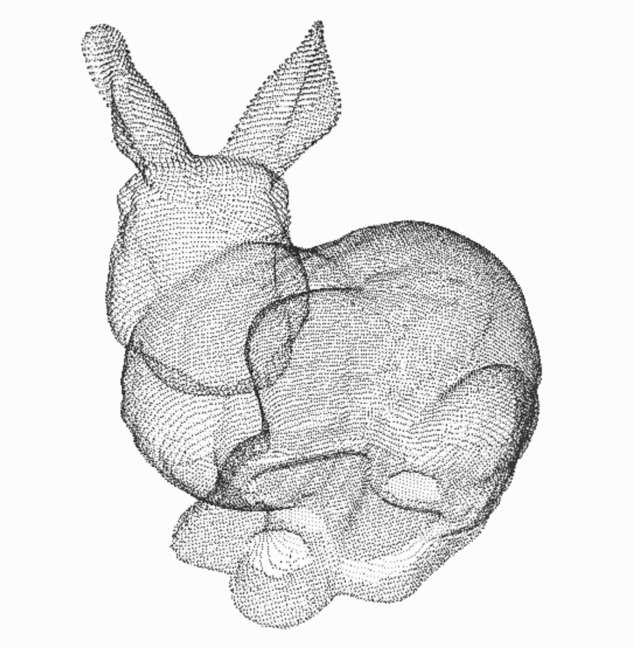


Implementation

- SparseConvNet
  - https://github.com/facebookresearch/SparseConvNet
  - Uses ResNet architecture
  - State-of-the-art for 3D analysis
  - Takes time to train

Graham et al., "**Submanifold Sparse Convolutional Networks**", *arxiv*

Point Networks



Point cloud

(The most common 3D sensor data)

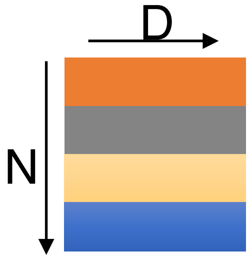
Directly Process Point Cloud Data

End-to-end learning for **unstructured, unordered** point data

Qi, Charles R., et al. "**Pointnet: Deep learning on point sets for 3d classification and segmentation**", CVPR 2017
Zaheer, Manzil, et al. "**Deep sets**", NeurIPS 2017

Properties of a Desired Point Network

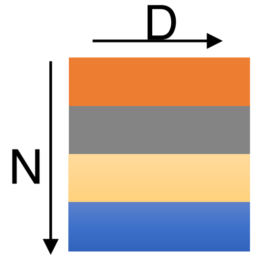
Point cloud: N **orderless** points, each represented by a D dim coordinate



2D array representation

Properties of a Desired Point Network

Point cloud: N **orderless** points, each represented by a D dim coordinate



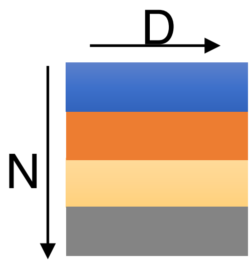
2D array representation

Permutation invariance

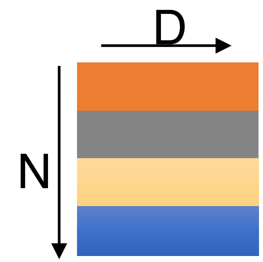
Transformation invariance

Permutation Invariance

Point cloud: N **orderless** points, each represented by a D dim coordinate



represents the same **set** as



2D array representation

Permutation Invariance: Symmetric Function

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

Examples:

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

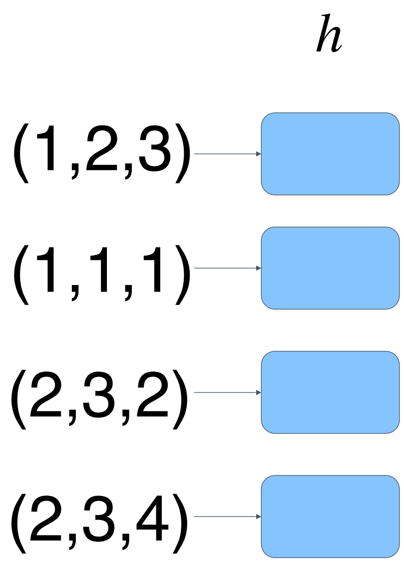
$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

…

Construct a Symmetric Function

Observe:

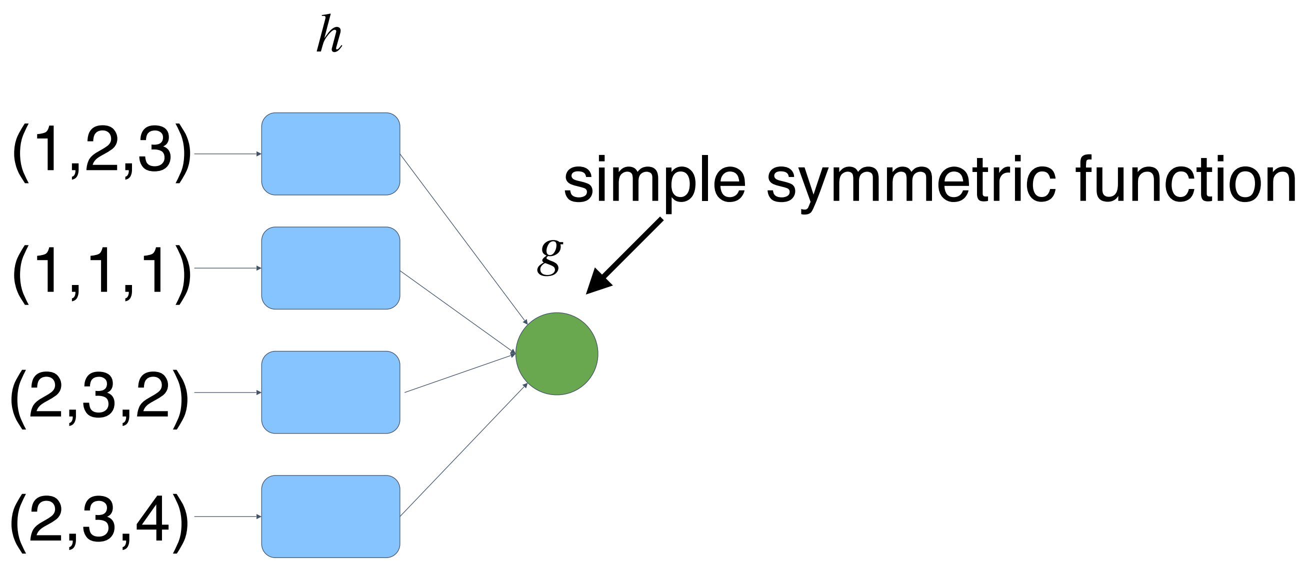
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Construct a Symmetric Function

Observe:

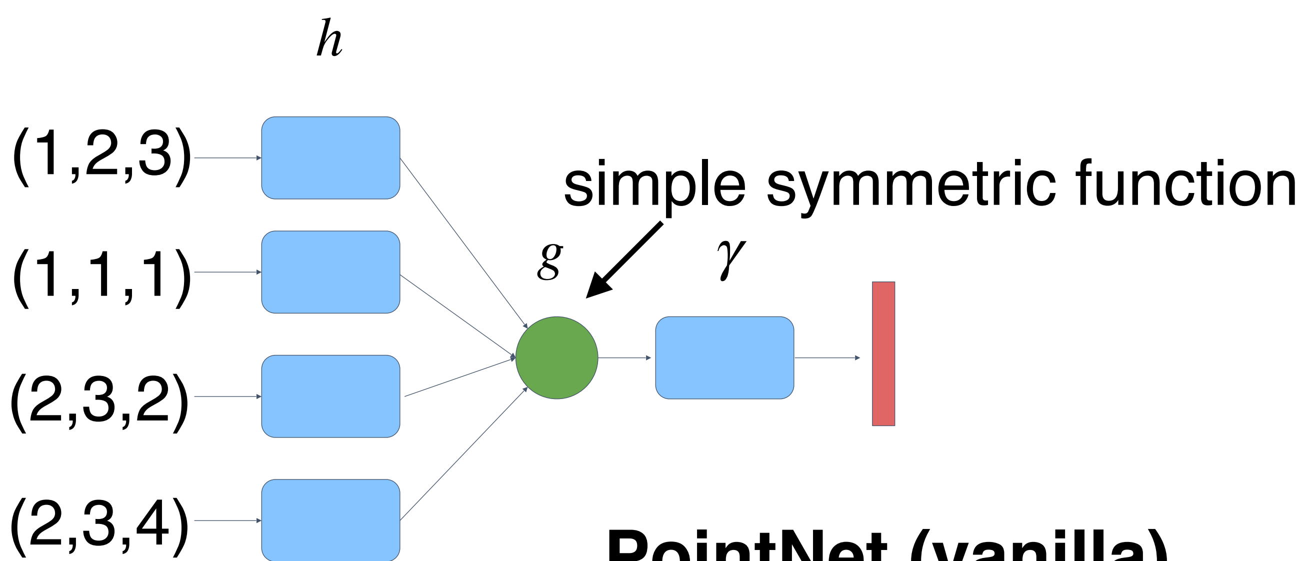
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Construct a Symmetric Function

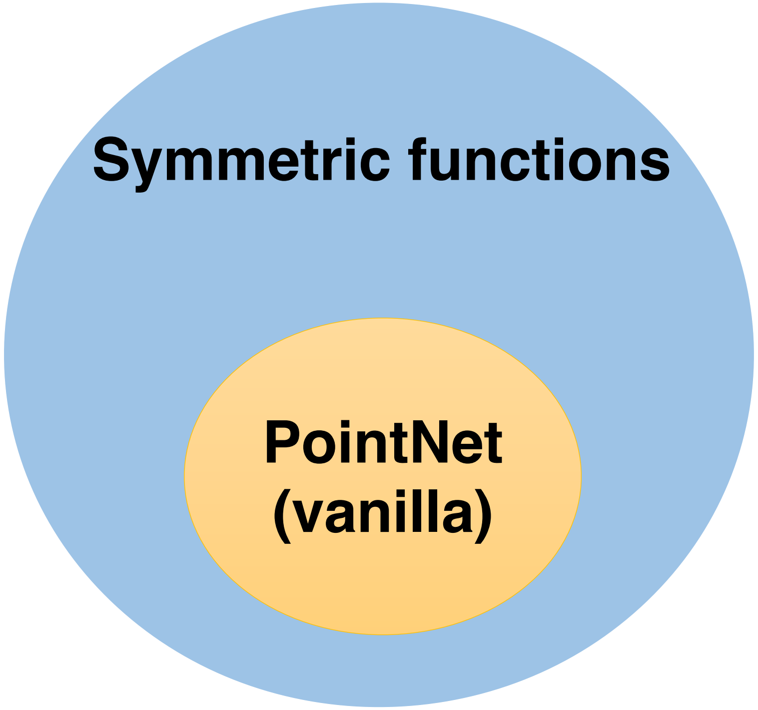
Observe:

$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



PointNet (vanilla)

Q: What Symmetric Functions Can Be Constructed by PointNet?



Universal Approximation Theorem

- Can approximate any “continuous” functions over sets
- “Continuous”: A function value would change by little if the point positions vary by little

$$\left| f(S) - \gamma\left(\underset{x_i \in S}{\mathrm{MAX}}\left\{h(x_i)\right\}\right)\right| < \epsilon$$

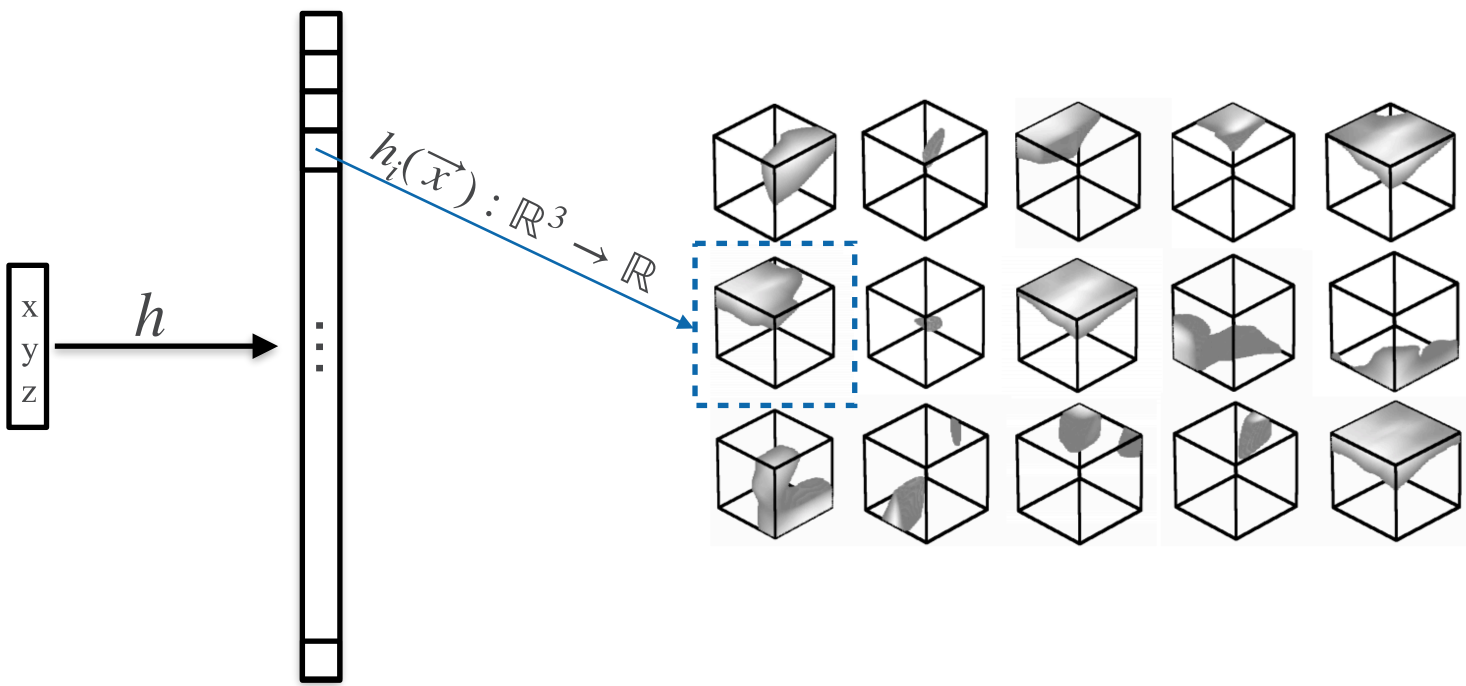
$S \subseteq \mathbb{R}^d,$

PointNet (vanilla)

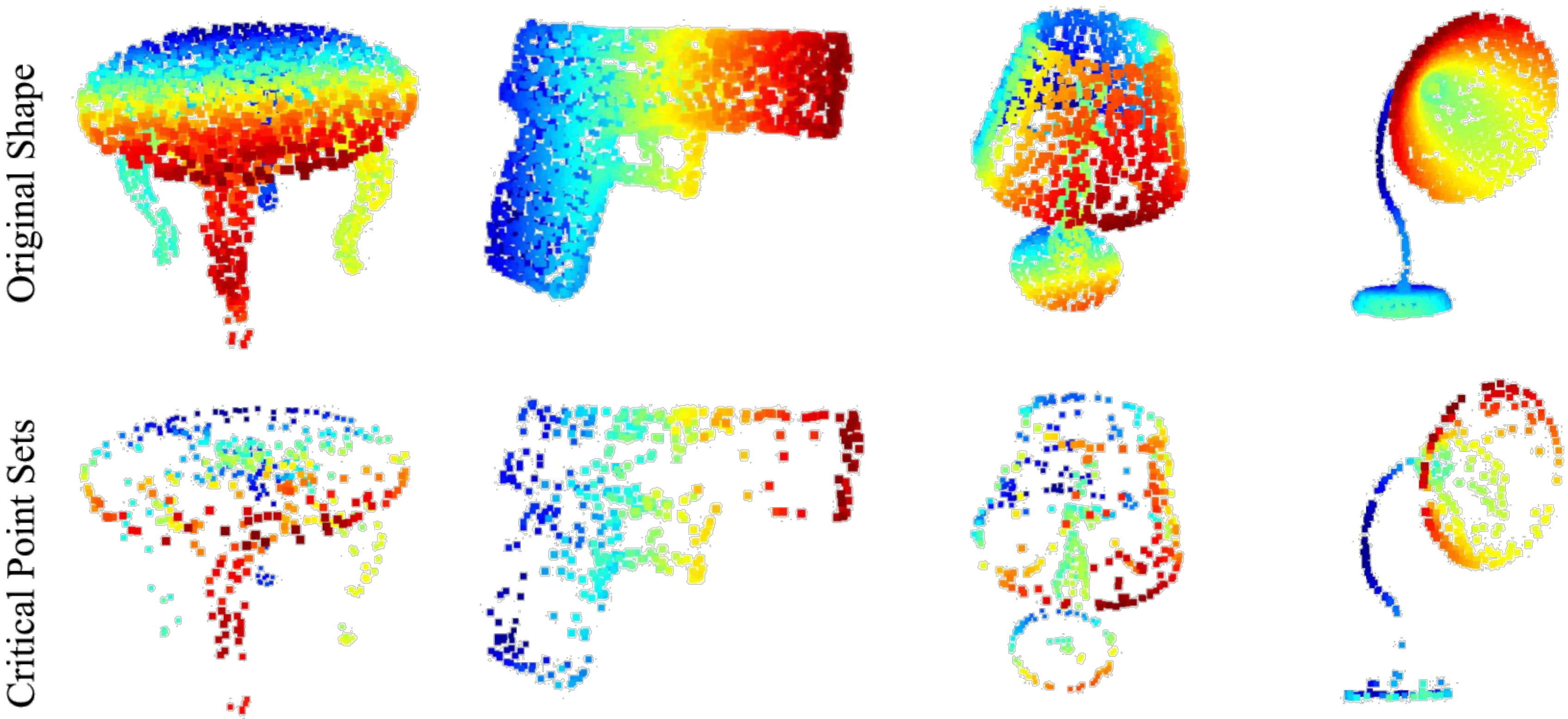
Interpretation to “First Layer” Output

- Think of each dimension as a “binary” variable (the truth is a soft version)
- It encodes whether the point is in a certain spatial region
- The shape of the spatial region is learned

3D voxels of irregular boundaries!



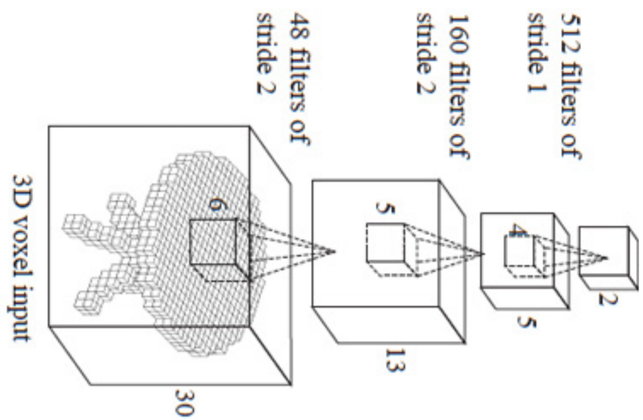
Salient Points: Points with Non-Zero Gradient w.r.t. Positions



Salient points are discovered!

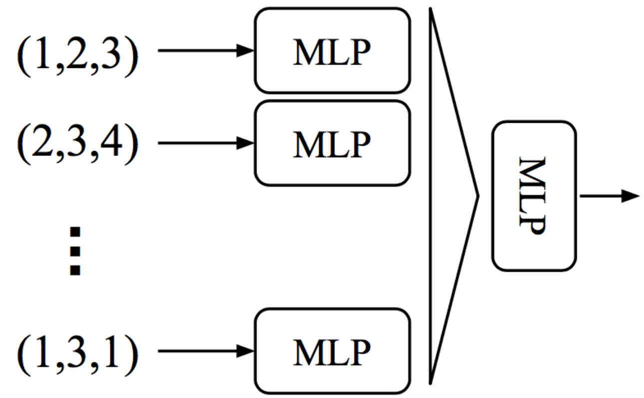
Limitations of PointNet

Hierarchical feature learning
Multiple levels of abstraction



3D CNN (Wu et al.)

Global feature learning
Either one point or all points



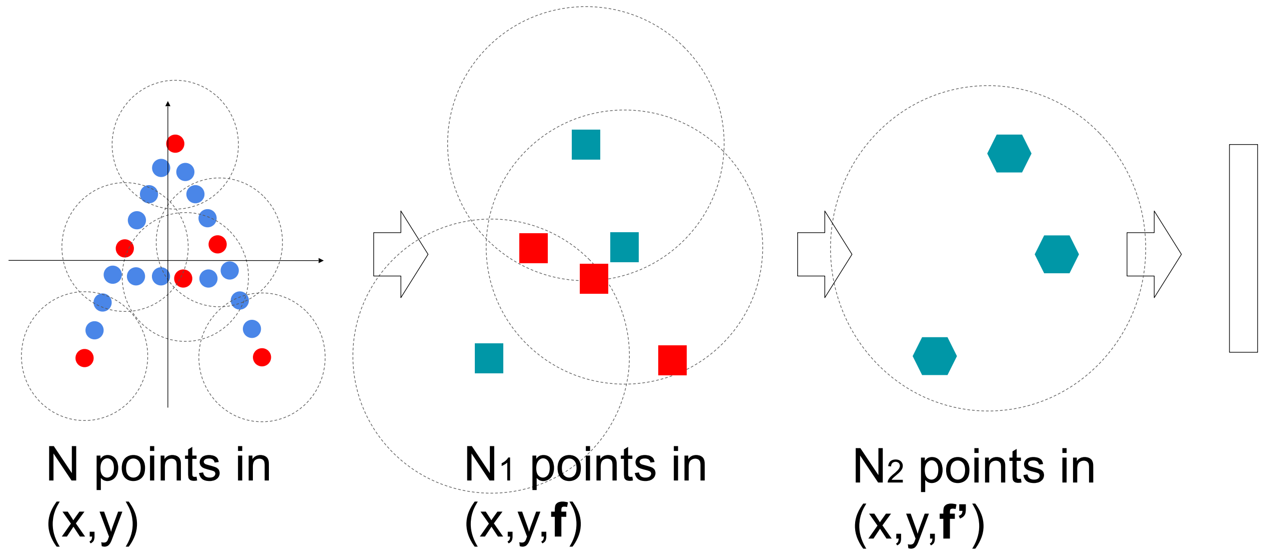
PointNet (vanilla) (Qi et al.)

- No local context for each point!
- Global feature depends on absolute coordinate. Hard to generalize to unseen scene configurations!

Points in Metric Space

- Learn "kernels" in 3D space and conduct convolution
- Kernels have compact spatial support
- For convolution, we need to find neighboring points
- Possible strategies for range query
  - Ball query (results in more stable features)
  - k-NN query (faster)

PointNet v2.0: Multi-Scale PointNet

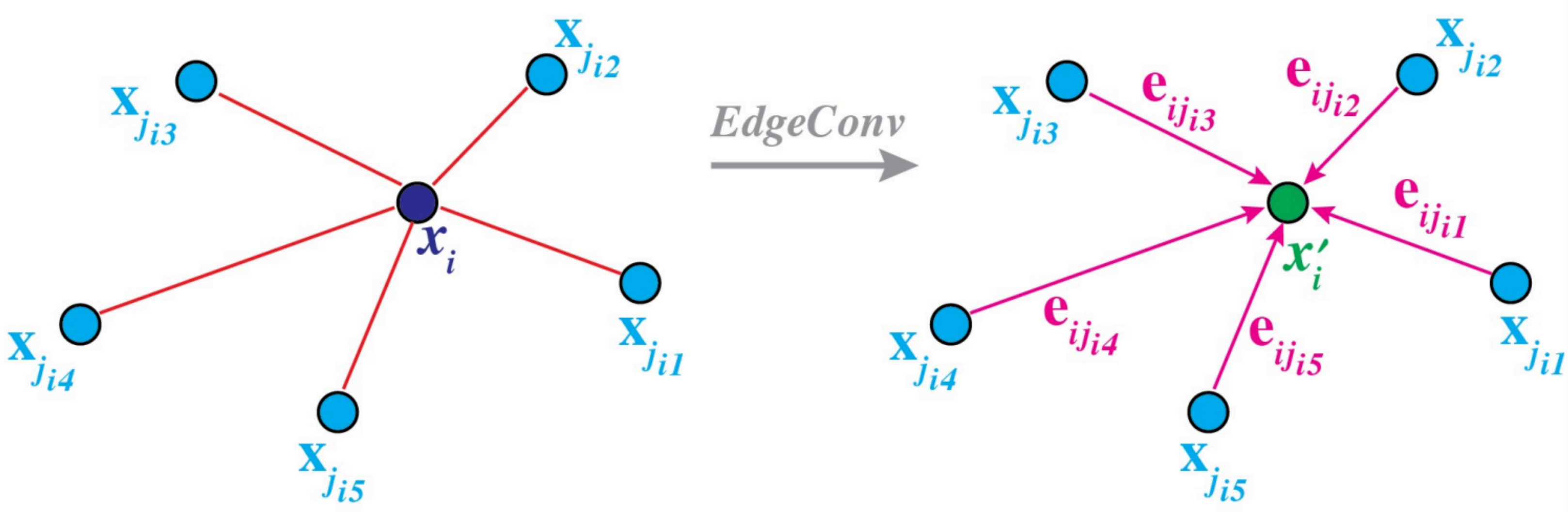


Repeat

- Sample anchor points
- Find neighborhood of anchor points
- Apply PointNet in each neighborhood to mimic convolution

Point Convolution As Graph Convolution

- Points -> Nodes
- Neighborhood -> Edges
- Graph CNN for point cloud processing



Wang et al., “**Dynamic Graph CNN for Learning on Point Clouds**”, *Transactions on Graphics, 2019*

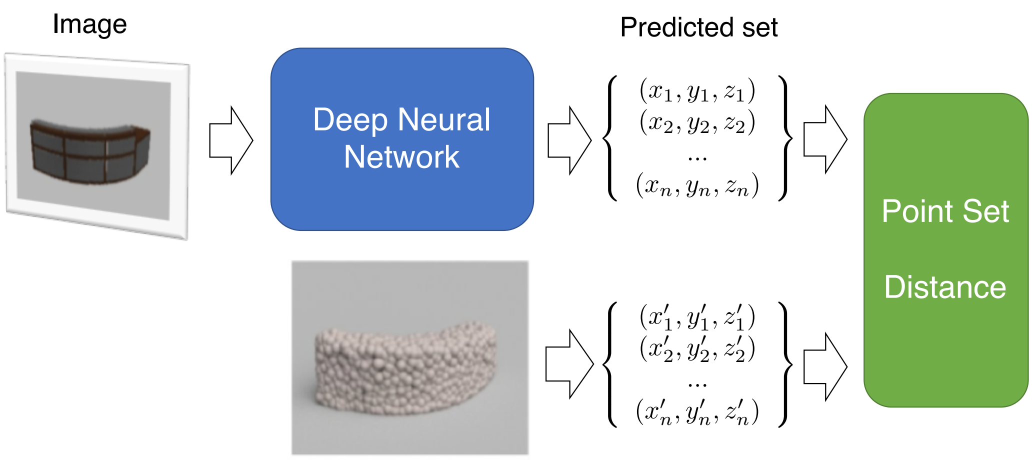
Liu et al., “**Relation-Shape Convolutional Neural Network for Point Cloud Analysis**”, *CVPR 2019*

Agenda

- 3D Classification
- **3D Reconstruction**

From Single Image to Point Cloud

- It is possible to generate a **set** (permutation invariant)



Fan et al., "**A Point Set Generation Network for 3D Object Reconstruction from a Single Image**", *CVPR 2017*