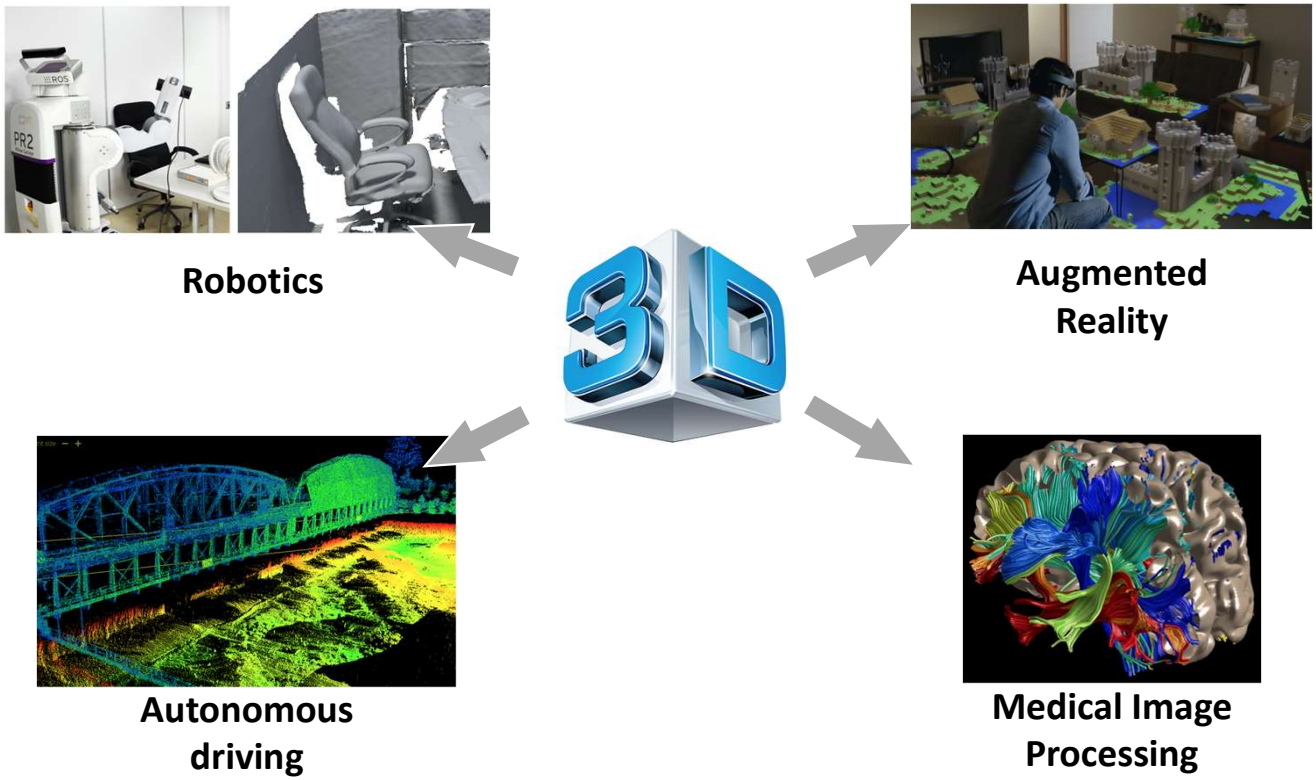
3D Deep Learning

QI CHEN
11/19/2020

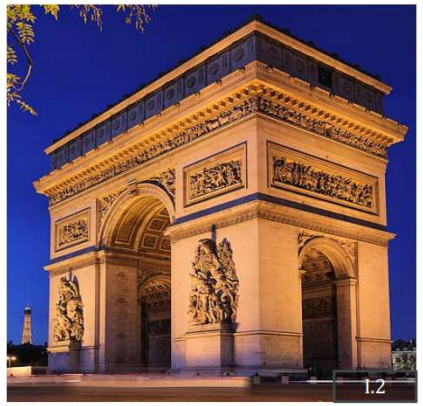
Broad Applications of 3D data



Traditional 3D Vision

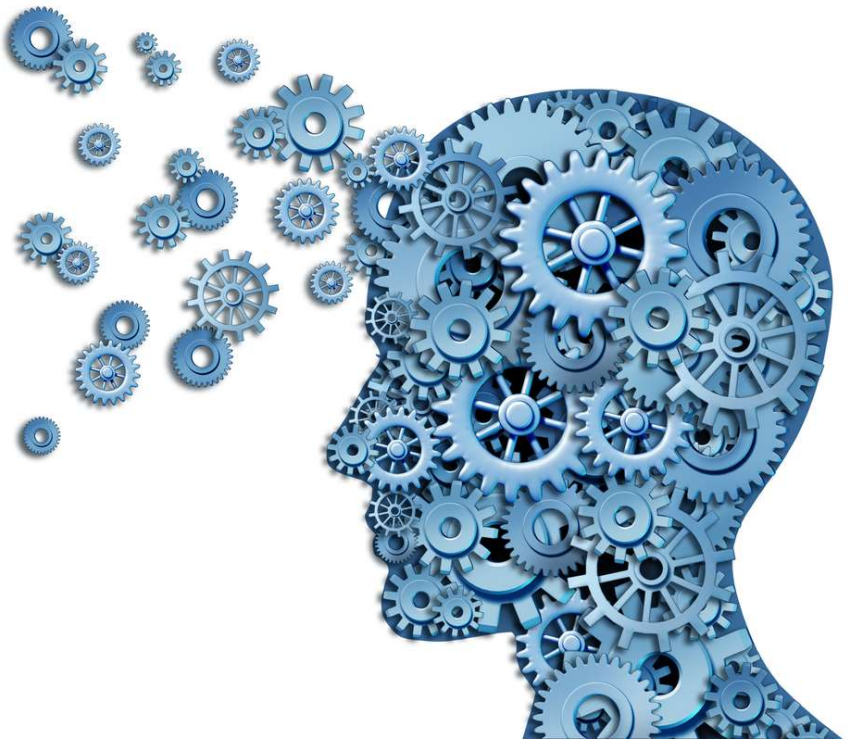
Multi-view Geometry: Physics based

Left Image

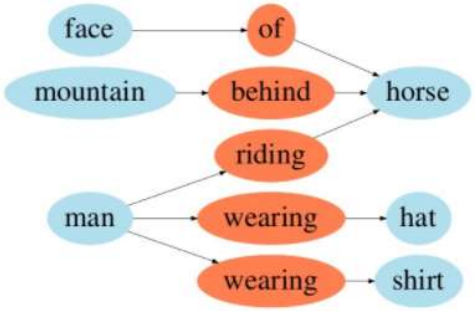


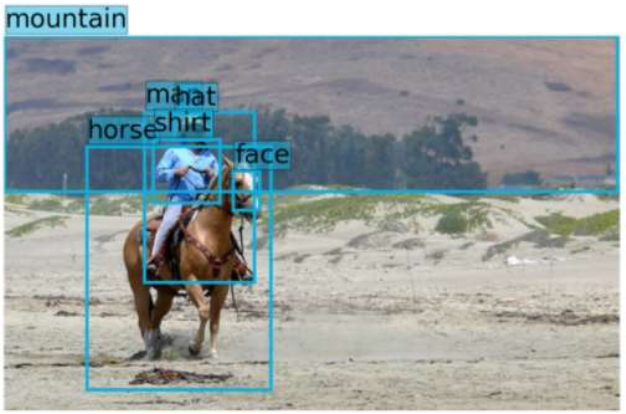
Right Image

3D Learning: Knowledge-based

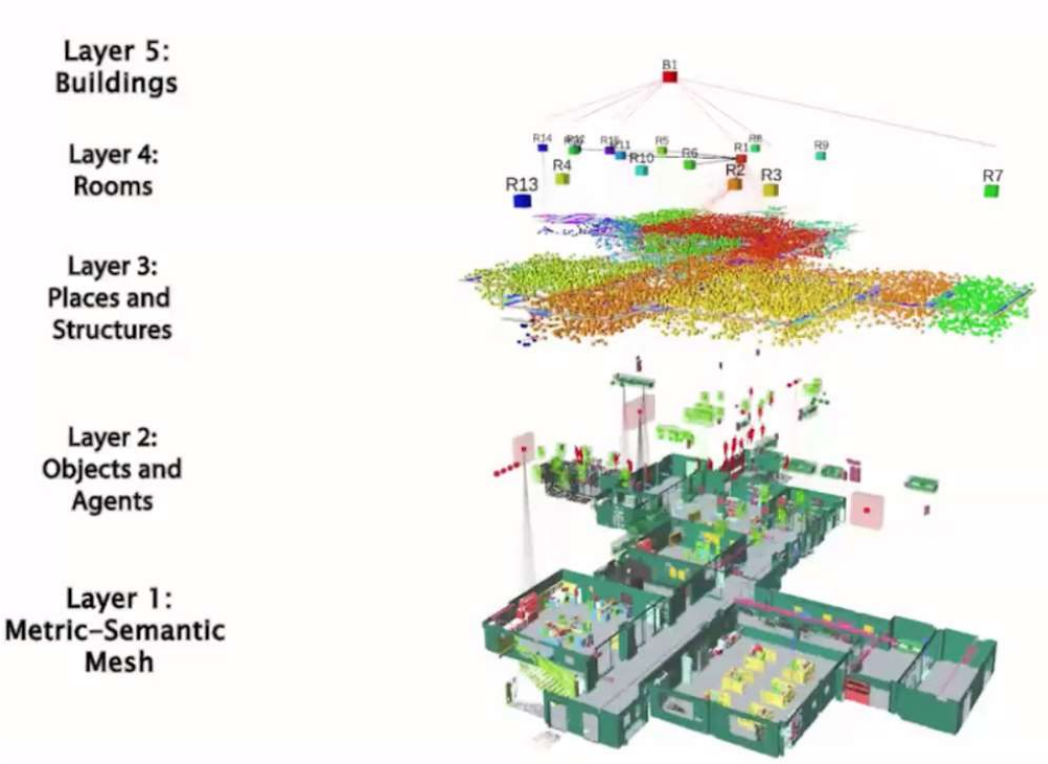


Holistic Scene Understanding: From 2D to 3D





Scene Graph



3D Scene Graph

https://visualgenome.org/
http://3dscenegraph.stanford.edu

Why 3D? - from a learning perspective

- 2D

Occlusion & Self-occlusion

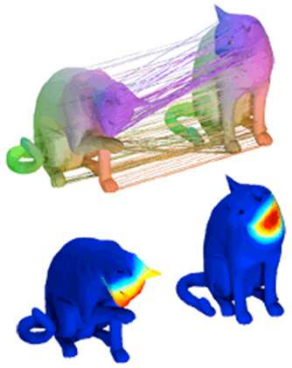
Deformation

View Angle Dependence

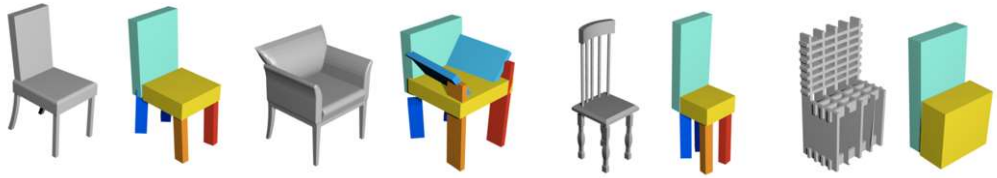
Holy Grail:
Less Data, More Reasoning

- 3D

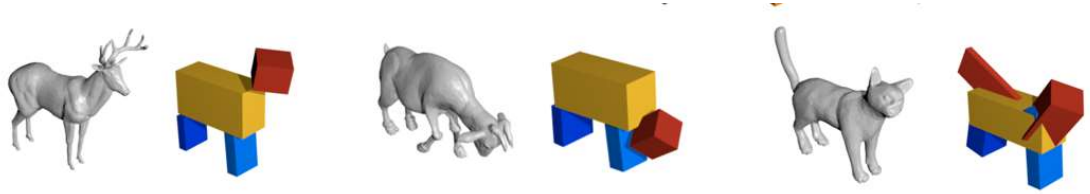
easier to be **related (correspondence)**



easier to be **compared**



easier to **abstracted**



Learning Objectives

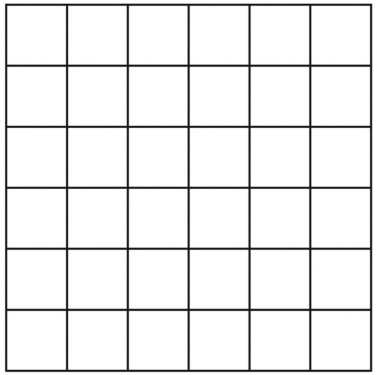
- Knowing Different 3D Data Representations
- Learning How to Process Irregular Data (Set, Graph)
- Understanding the Design Principal of Local Feature Extraction
- Understanding Pros and Cons of Grid CNN, PointNet and Graph CNN

Topics

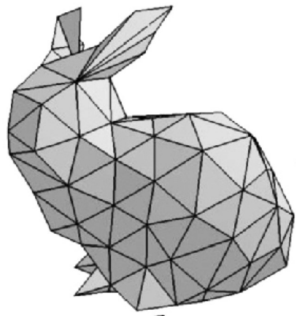
- **3D Data**
- Models (mainly classification)

3D Data Representation

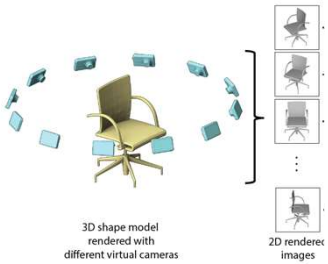
Rasterized form (regular grids)



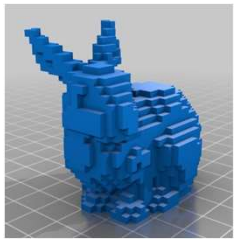
Geometric form (irregular)



3D Data Representation

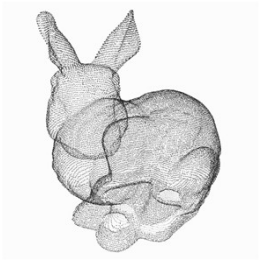


Multi-view

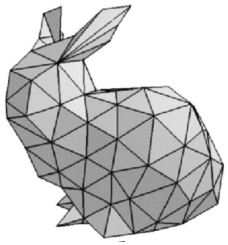


Volumetric

Part Assembly



Point Cloud



Mesh

$$F(x) = 0$$

Implicit Shape

Datasets for 3D Objects

Large-scale Synthetic Objects: ShapeNet

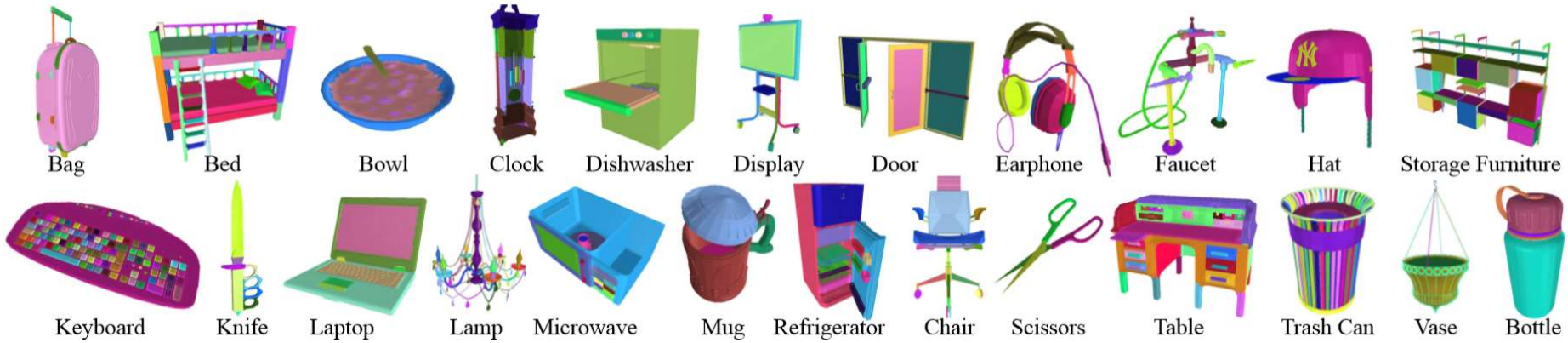
3DScan: Consumer-grade 3D scanning

Chang et al., “**ShapeNet: An Information-Rich 3D Model Repository**” , *arXiv*
Wu et al., "**3D ShapeNets: A deep representation for volumetric shapes**", *CVPR 2015*
Choi et al., “**A Large Dataset of Object Scans**", *arXiv*

Datasets for 3D Object Parts

Fine-grained Part: PartNet (ShapeNetPart2019)

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*

Datasets for Indoor 3D Scenes

Large-scale Synthetic Scenes: SceneNet

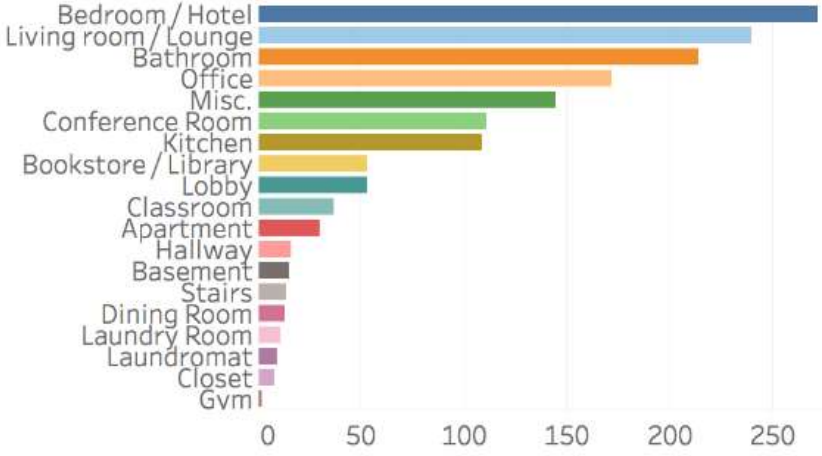
3D meshes
5M Photorealistic Images

Ankur et al., "**Understanding RealWorld Indoor Scenes with Synthetic Data**", *CVPR 2016*
McCormac et al., "**SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?**", *ICCV 2017*

Datasets for Indoor 3D Scenes

Large-scale Scanned Real Scenes: ScanNet

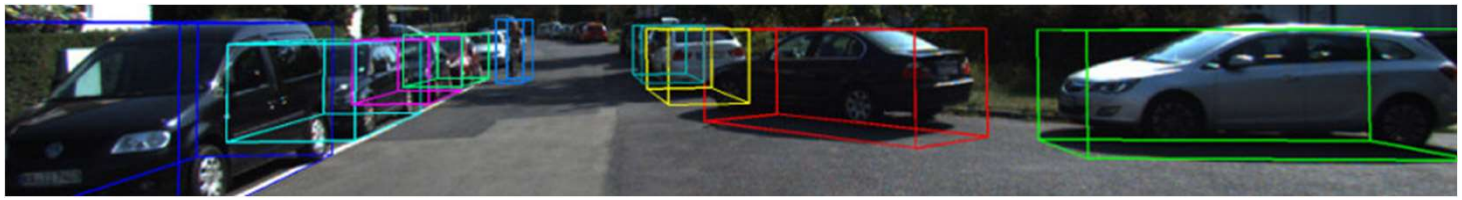
2.5 M Views in 1500 RGBD scans
3D camera poses
surface reconstructions
Instance-level semantic segmentations



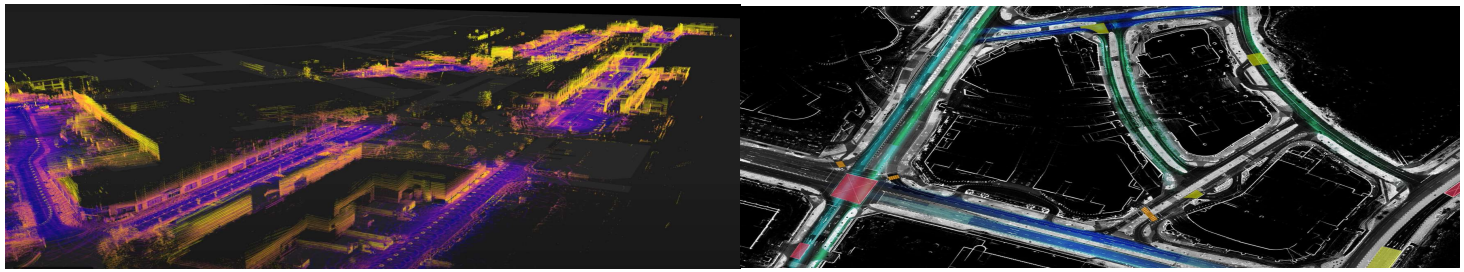
Dai et al., "**ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes**", *CVPR 2017*

Datasets for Outdoor 3D Scenes

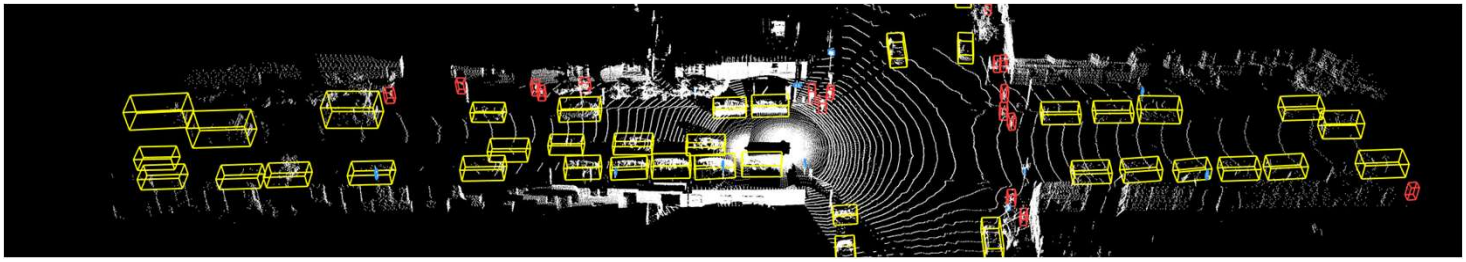
KITTI: LiDAR data, labeled by 3D bounding boxes



NuScenes: LiDAR data, labeled per point & by 3D bounding boxes, HD Map included



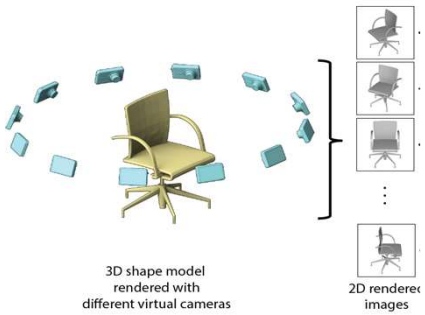
Waymo Open Dataset: LiDAR data, labeled by 3D bounding boxes



Topics

- 3D Data
- **Models (mainly classification)**

Data



Multi-view

Model

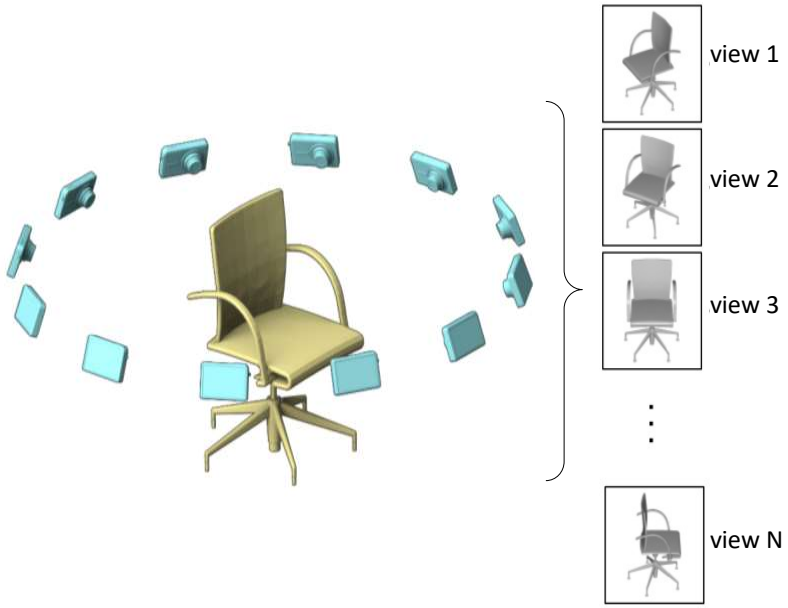
Multi-View CNN

Given an Input Shape

This is a chair

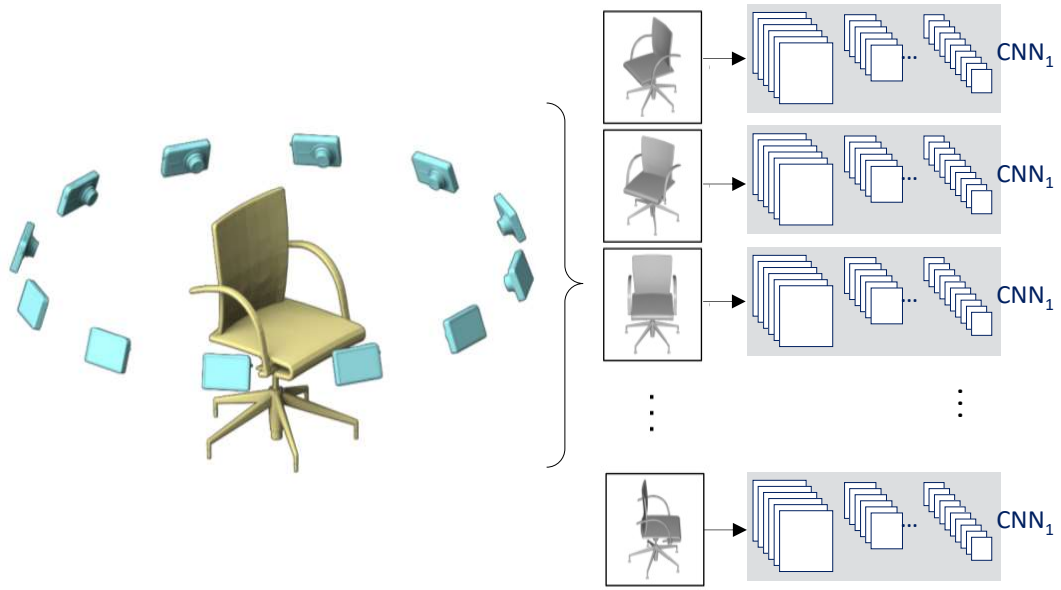
Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

Render with Multiple Virtual Cameras



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

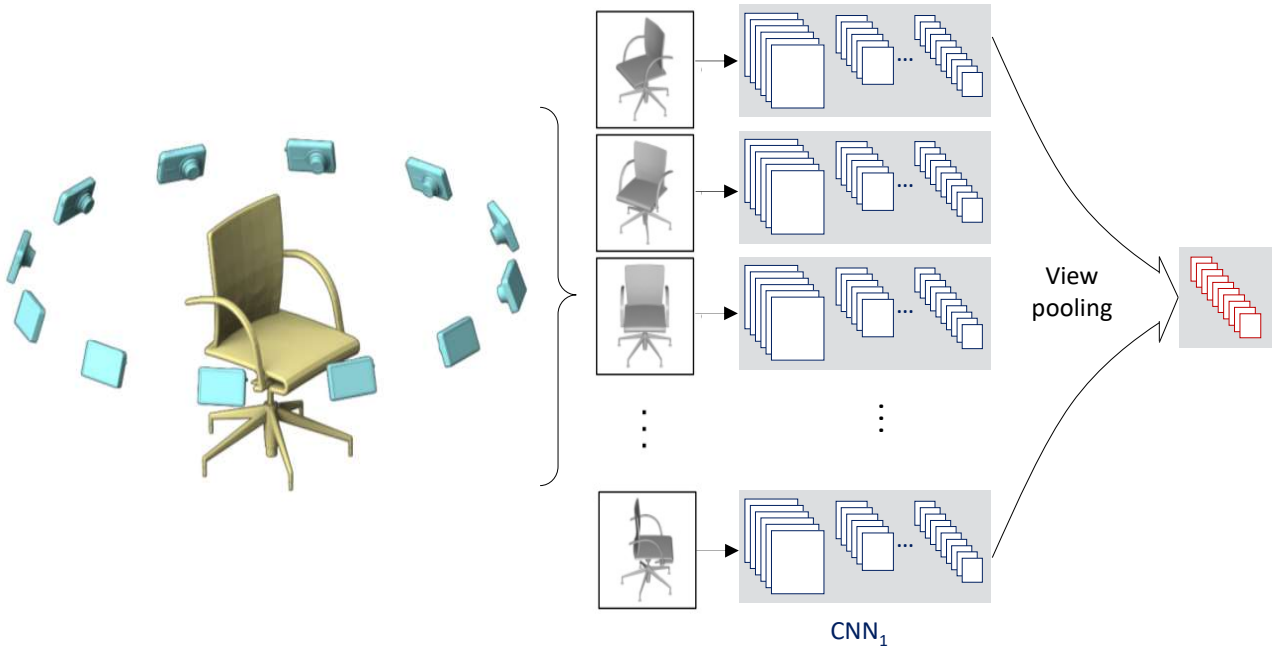
The Rendered Images are Passed through CNN_1 for Image Features



CNN_1: a ConvNet extracting image features

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

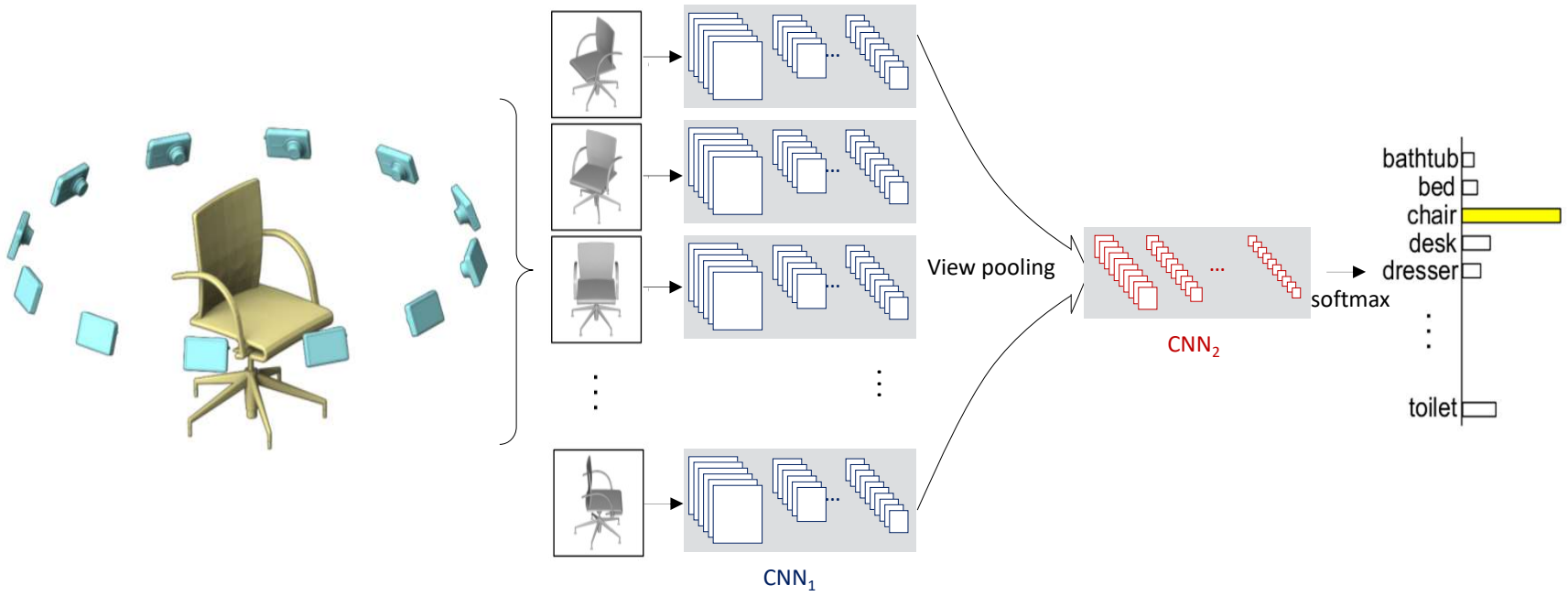
All Image Features are Combined by View Pooling



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



CNN_2: a second ConvNet producing shape descriptors

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

Experiments – Classification & Retrieval

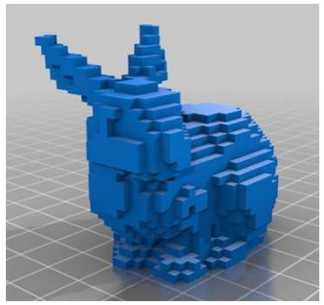
	Method	Classification (Accuracy)	Retrieval (mAP)
Non-deep	SPH [16]	68.2%	33.3%
	LFD [5]	75.5%	40.9%
	3D ShapeNets [37]	77.3%	49.2%
	FV, 12 views	84.8%	43.9%
	CNN, 12 views	88.6%	62.8%
	MVCNN, 12 views	**89.9%**	70.1%
	MVCNN+metric, 12 views	89.5%	**80.2%**
	MVCNN, 80 views	90.1%	70.4%
	MVCNN+metric, 80 views	**90.1%**	**79.5%**

On ModelNet40

- ✅ Indeed gives good performance
- ✅ Can leverage vast literature of image classification
- ✅ Can use pretrained features
- ❌ Need projection

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

Data



Volumetric

Model

3D CNNs

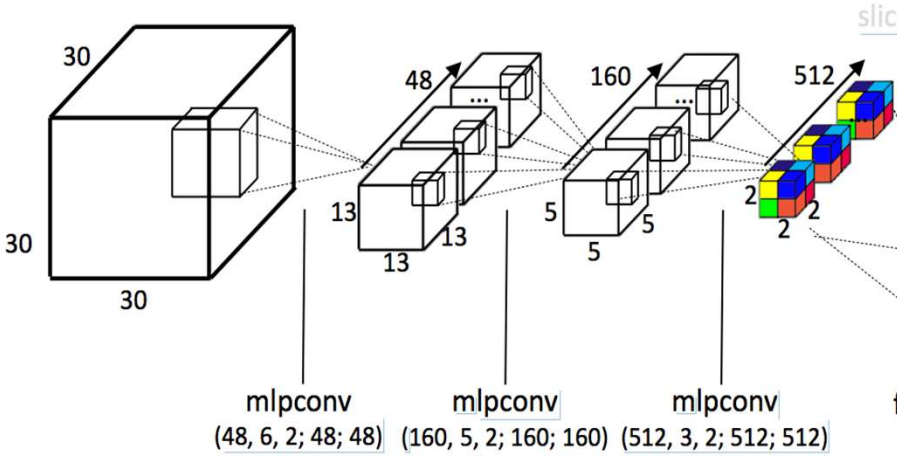
Voxelization into Occupancy Grids

Represent the occupancy of regular 3D grids

1. Divide 3D space into uniform 3D grids (cuboid-shape voxels)
2. Voxel value is 1 if it contains points/is solid and 0 otherwise
   non-empty/solid voxels vs empty/null voxels

3D CNN on Volumetric Data

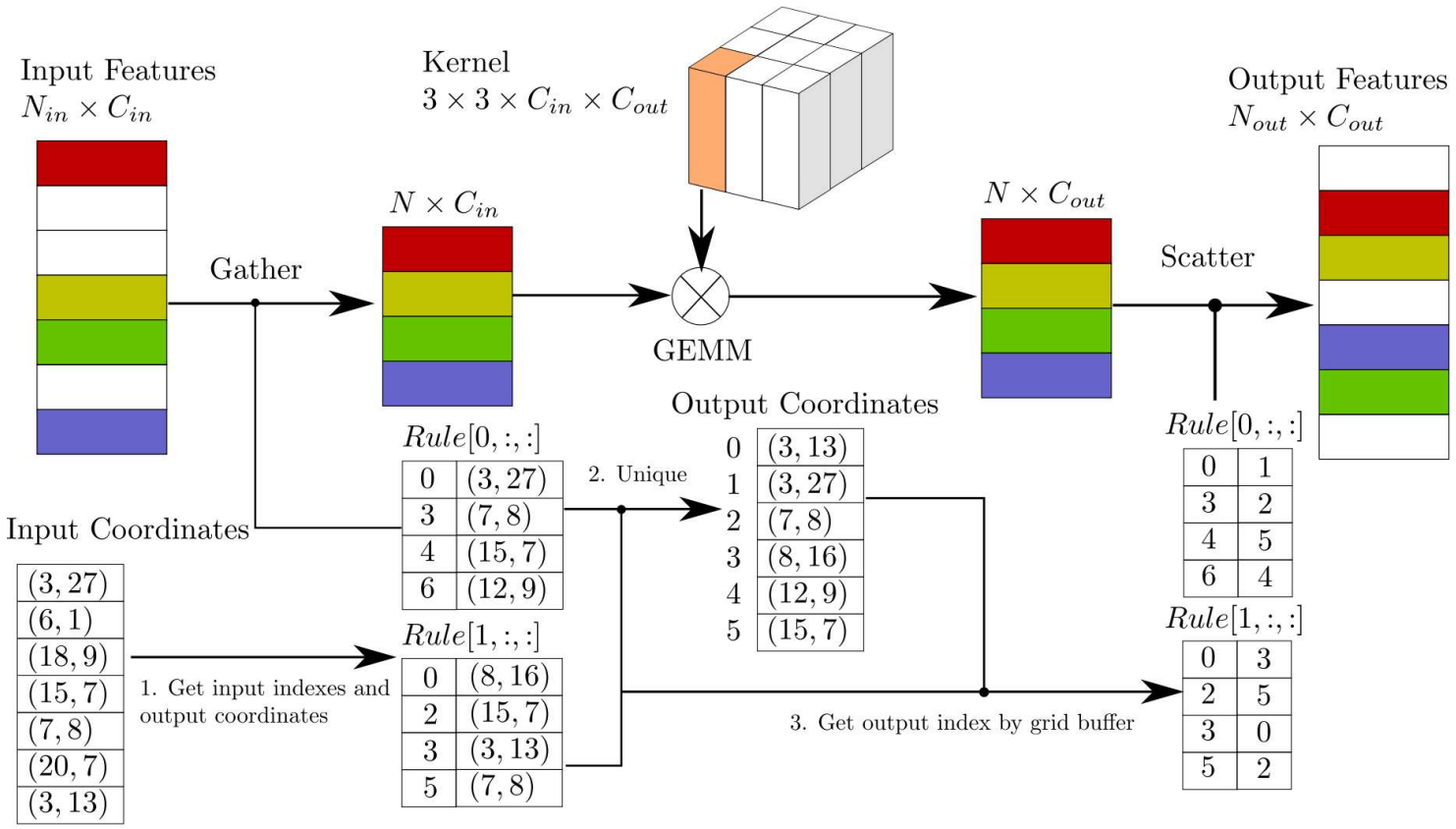
3D convolution uses 4D kernels



Issue: Data Sparseness -> Memory Inefficient

- Most voxels have zero values – sparse
- In auto-driving scenes, only ~0.07% of voxels are non-empty

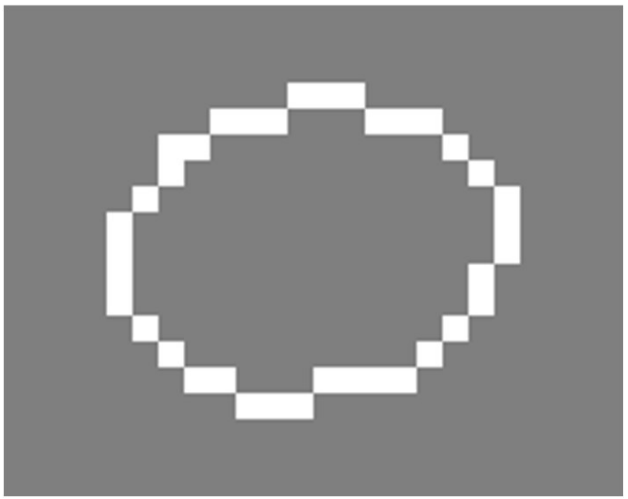
Data Sparseness -> Sparse Convolution



https://github.com/facebookresearch/SparseConvNet

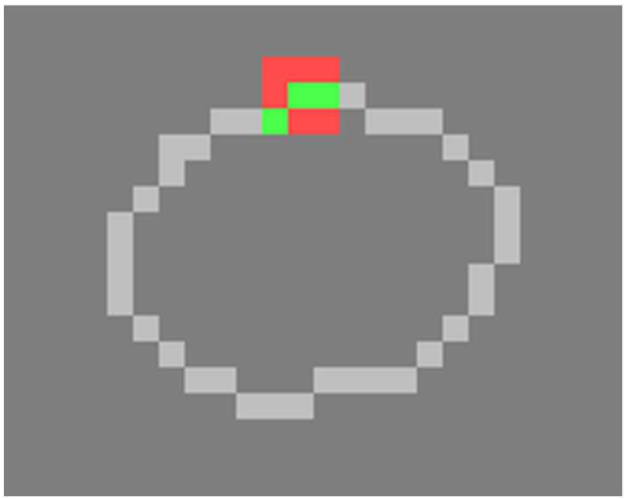
Data Sparseness -> Sparse Convolution

- Submanifold Sparse Convolution



Regular (Sparse) Conv

A 3x3 kernel would spread non-zeros values to neighbors



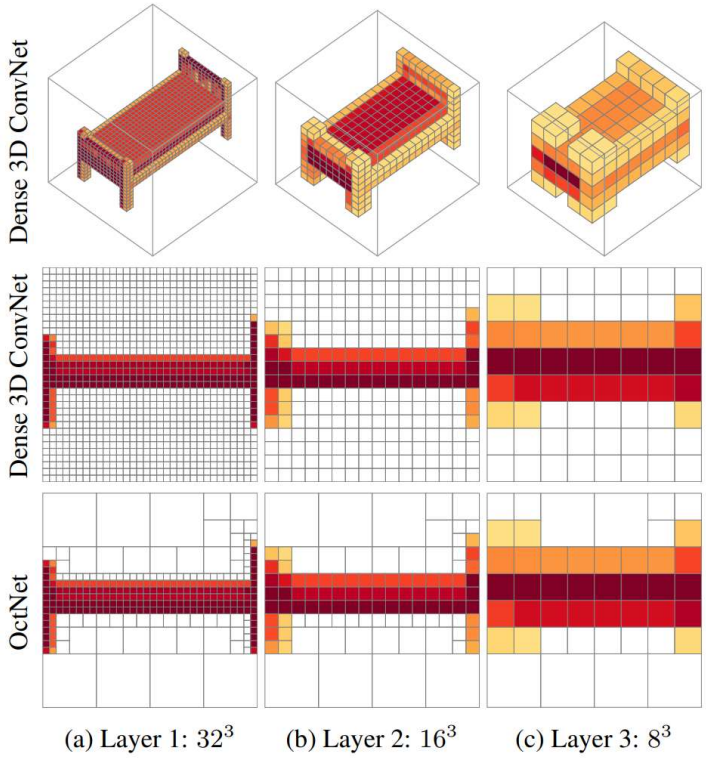
Submanifold Sparse Conv

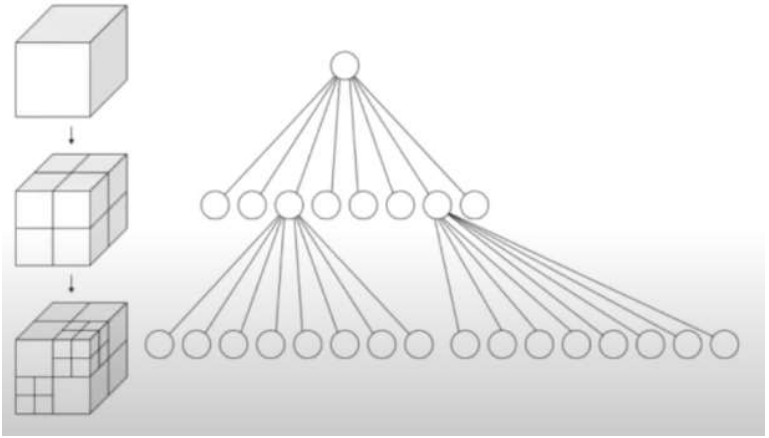
Output is non-zero only if its corresponding input position has non-zero values. sparsity-preserving

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

Data Sparseness -> Oct Tree

- OctNet

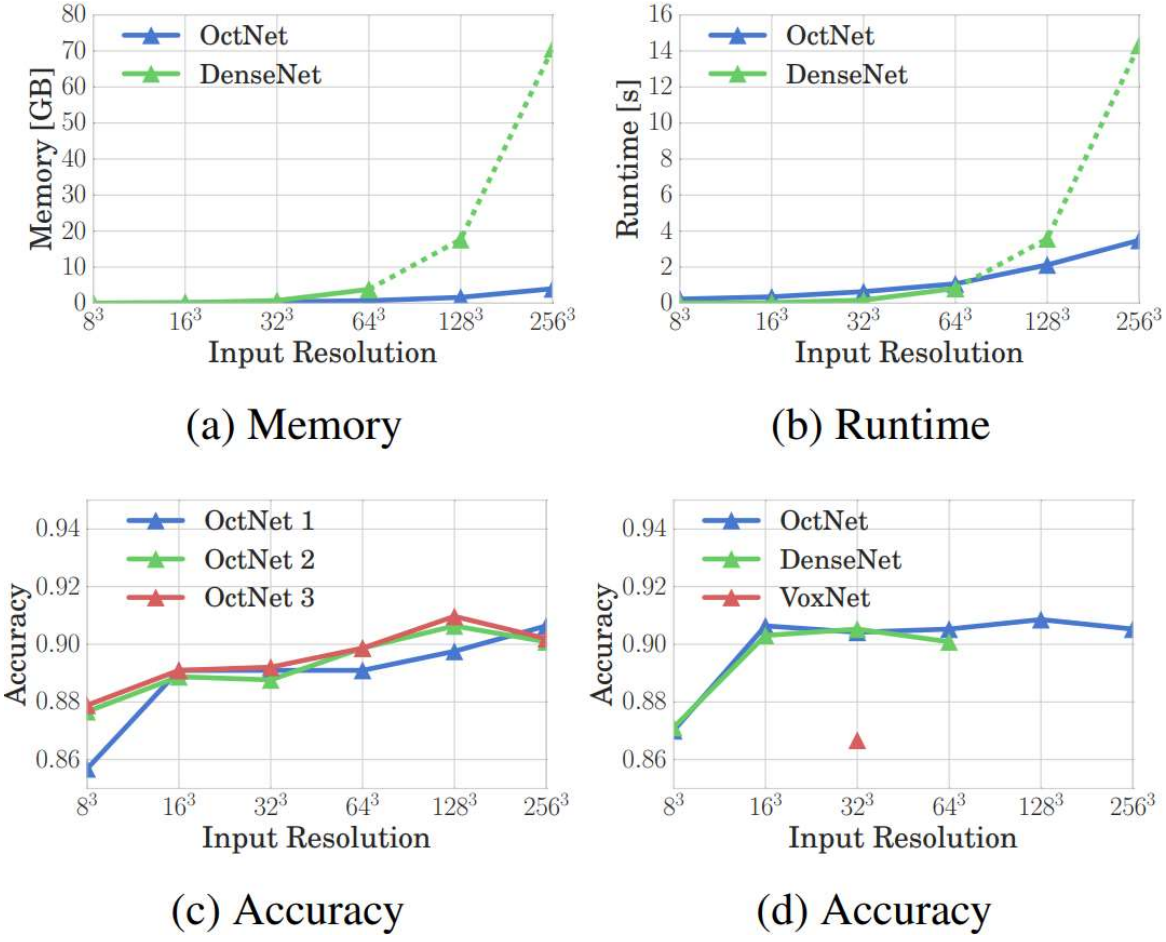




- Need to redefine operations such as convolution and pooling
- Unary/In-place operations remain the same
- Heavy Engineering, not so popular

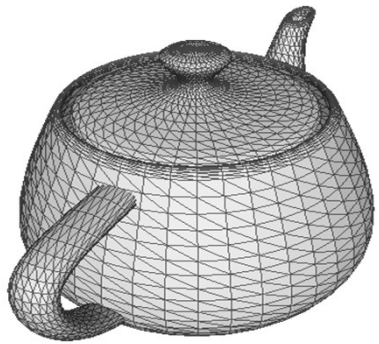
Riegler et al., "**OctNet: Learning Deep 3D Representations at High Resolutions**", *CVPR 2017*

Results on ModelNet10 Classification Task

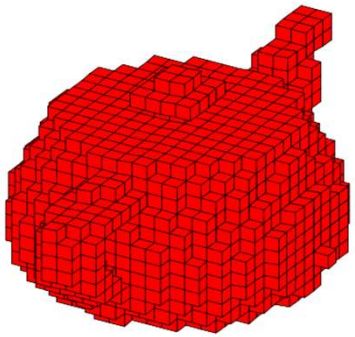


OctNet enables higher input resolution under centain memory limit

Issue: Quantization -> Loss of Information



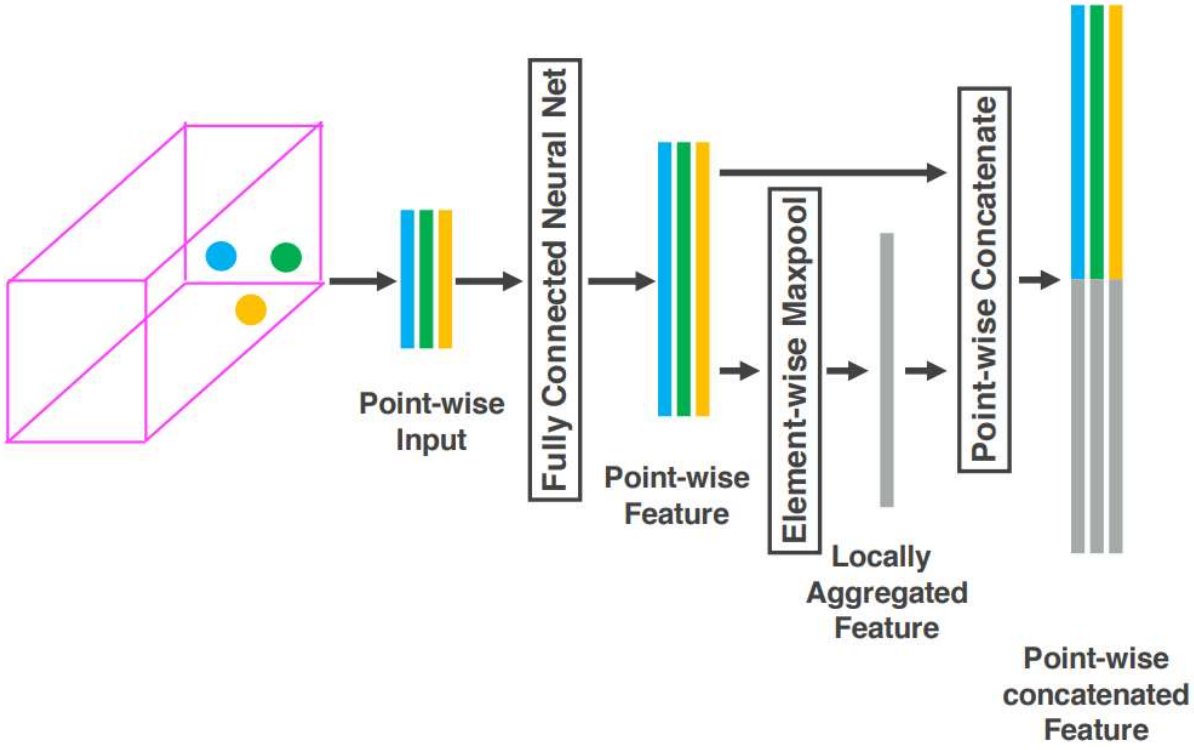
Polygon Mesh



Occupancy Grid
30x30x30

Quantization -> Continuous Voxel Features

- VoxelNet – Voxel Feature Encoder



Zhou et al., "**VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection**", *CVPR 2018*

Recap

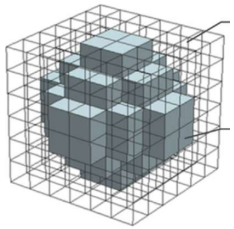
- Regular (Grid-Structure) Data: from 2D to 3D

2D	3D
Pixel	Voxel
Pixel values: RGB	Voxel Values: 0/1 occupancy; Continuous Features from Voxel Feature Encoder
2D CNN	3D CNN
2D images are compact	3D data is large and sparse

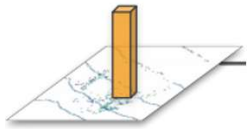
To save memory:
(Submanifold) Sparse Convolution
Oct Tree

Both have trade-offs between accuracy and pixel/voxel resolution (constrained by memory) !

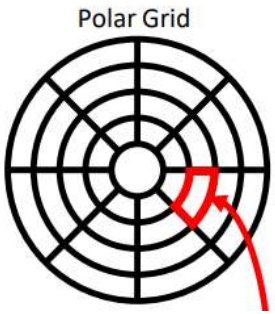
Extension: Different Voxel Shapes

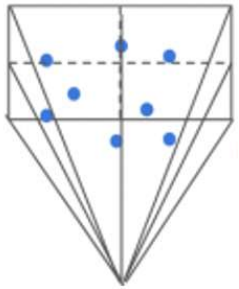


Cuboid

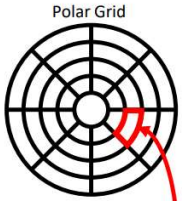


Pillar

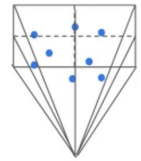




Bird Eye's View:



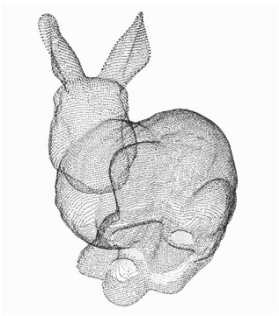
Range View: (front perspective)



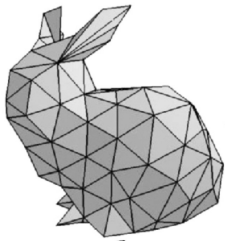
	Cartesian	Cylindrical	Spherical	HCS
3D Voxel	$(x, y, z, \Delta x, \Delta y, \Delta z)$	$(r, \theta, z, \Delta r, \Delta\theta, \Delta z)$	$(R, \theta, \phi, \Delta R, \Delta\theta, \Delta\phi)$	$(r, \theta, \phi, \Delta r, \Delta\theta, \Delta\phi)$
BEV Voxel	$(x, y, \Delta x, \Delta y)$	$(r, \theta, \Delta r, \Delta\theta)$	N/A	$(r, \theta, \Delta r, \Delta\theta)$
RV Voxel	N/A	N/A	$(\theta, \phi, \Delta\theta, \Delta\phi)$	$(\theta, \phi, \Delta\theta, \Delta\phi)$
Details		$r = \sqrt{x^2 + y^2}$ $\theta = \arctan y/x$	$R = \sqrt{x^2 + y^2 + z^2}$ $\theta = \arctan y/x$ $\phi = \arccos z/R$	$r = \sqrt{x^2 + y^2}$ $\theta = \arctan y/x$ $\phi = \arctan z/r$
References	VoxelNet [1], MVF [14]	Alsfasser et al [12]	MVF [14]	Ours

Lang et al., "**PointPillars: Fast Encoders for Object Detection from Point Clouds**", *CVPR 2019*
*Zhou et al., "***End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds** ", CoRL 2019
Alsfasser et al., "**Exploiting polar grid structure and object shadows for fast object detection in point clouds**", ICMV 2020
Chen et al., "**Every View Counts: Cross-View Consistency in 3D Object Detection with Hybrid-Cylindrical-Spherical Voxelization**", NeurIPS 2020

Data



Point cloud
(The most common 3D sensor data)



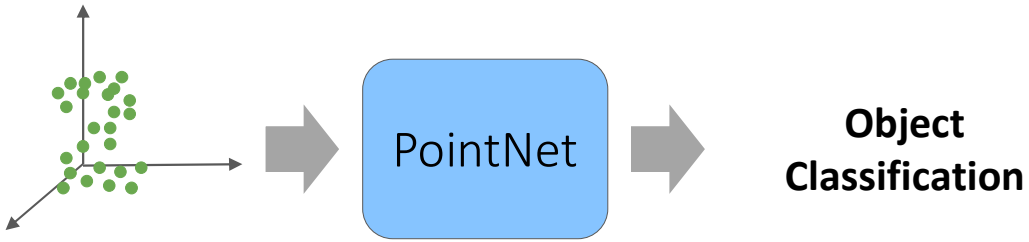
Mesh

Model

PointNet
Graph CNN

Directly Process Point Cloud Data

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



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

Zaheer et al. "**Deep sets**", NeurIPS 2017

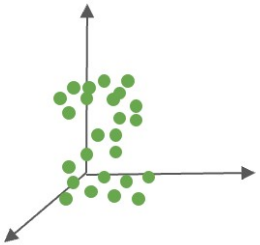
3D Points

(1,2,3) →

(1,1,1) →

(2,3,2) →

(2,3,4) →



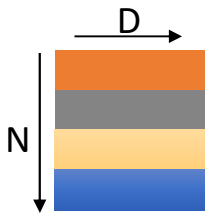
Which of the following operations can be applied to these points?

- ✗ Regular Convolution
- ✓ Fully Connected/Dense Layer
- ✓ Pooling (Max, Avg)
- ✓ In-place Operations (ReLu, Sigmoid)
- ✓ 1x1 Convolution

CNNs cannot be applied to irregular/unstructured data!

Permutation invariance

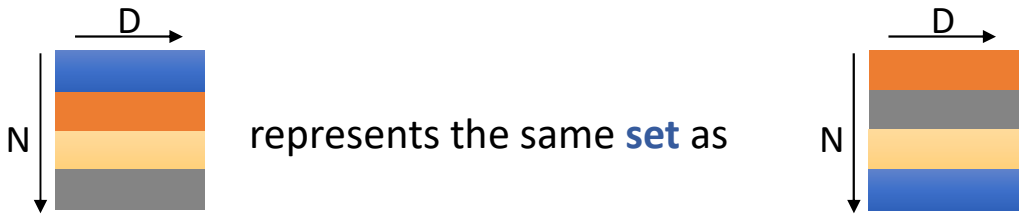
Point cloud: N **orderless** points, each represented by a D dim coordinate



2D array representation

Permutation invariance

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

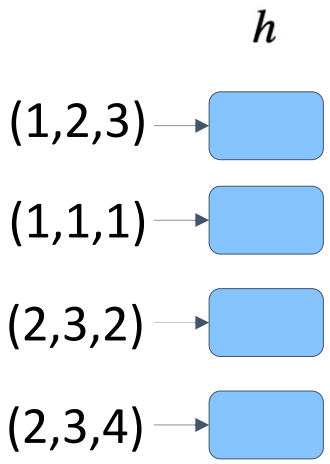


2D array representation

Construct a Symmetric Function

Observe:

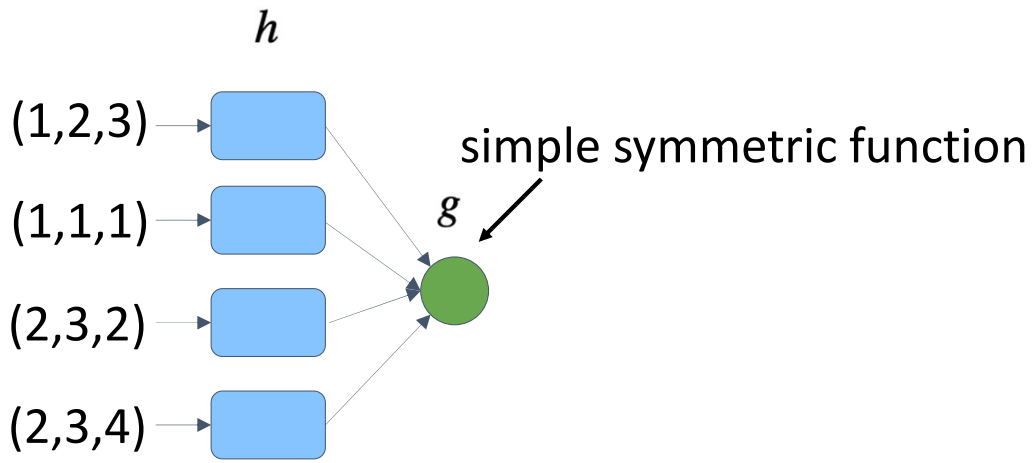
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



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



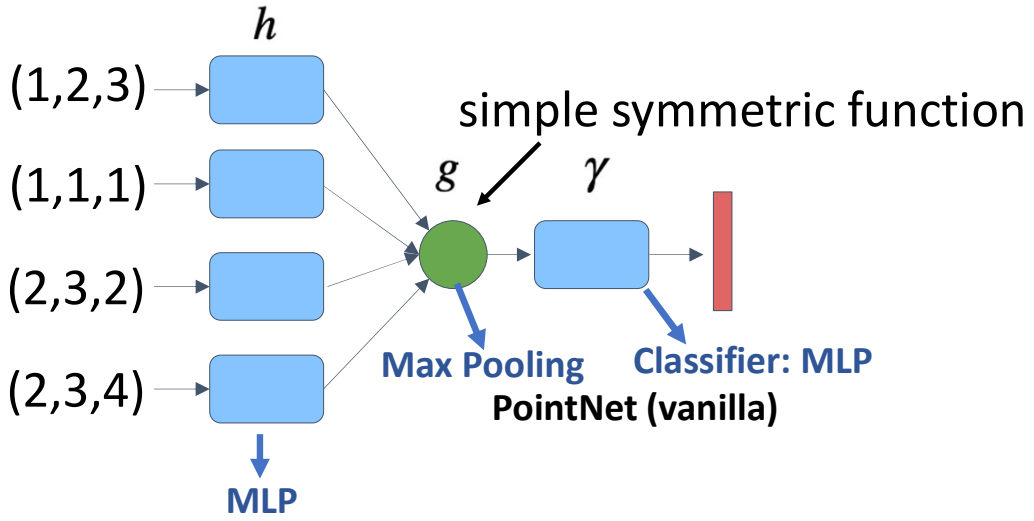
Which of the followings are symmetric functions?

- ✗ f(x1, x2) = x1
- ✗ f(x1, x2) = w1*x1+w2*x2 — **Fully connected layer**
- ✓ f(x1, x2) = 1
- ✓ f(x1, …, xn) = mean(x1, …, xn) — **Avg Pooling**
- ✓ f(x1, …, xn) = max(x1, …, xn) — **Max Pooling**

Construct a Symmetric Function

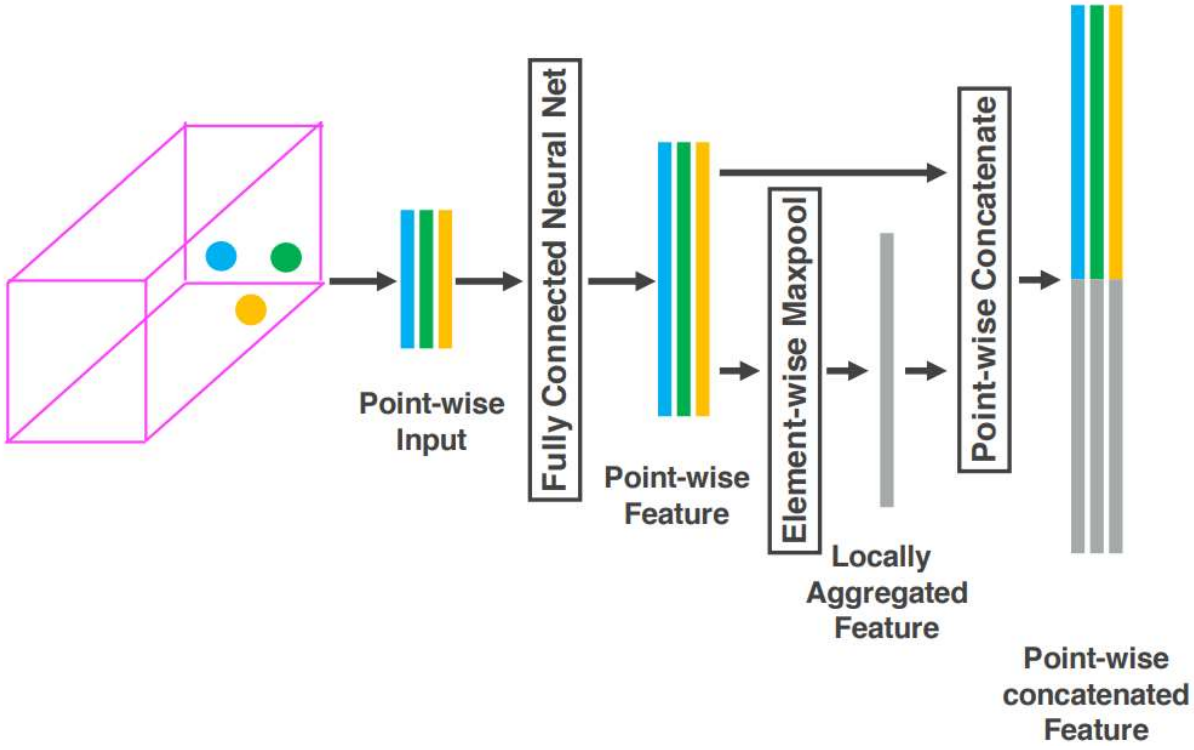
Observe:

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



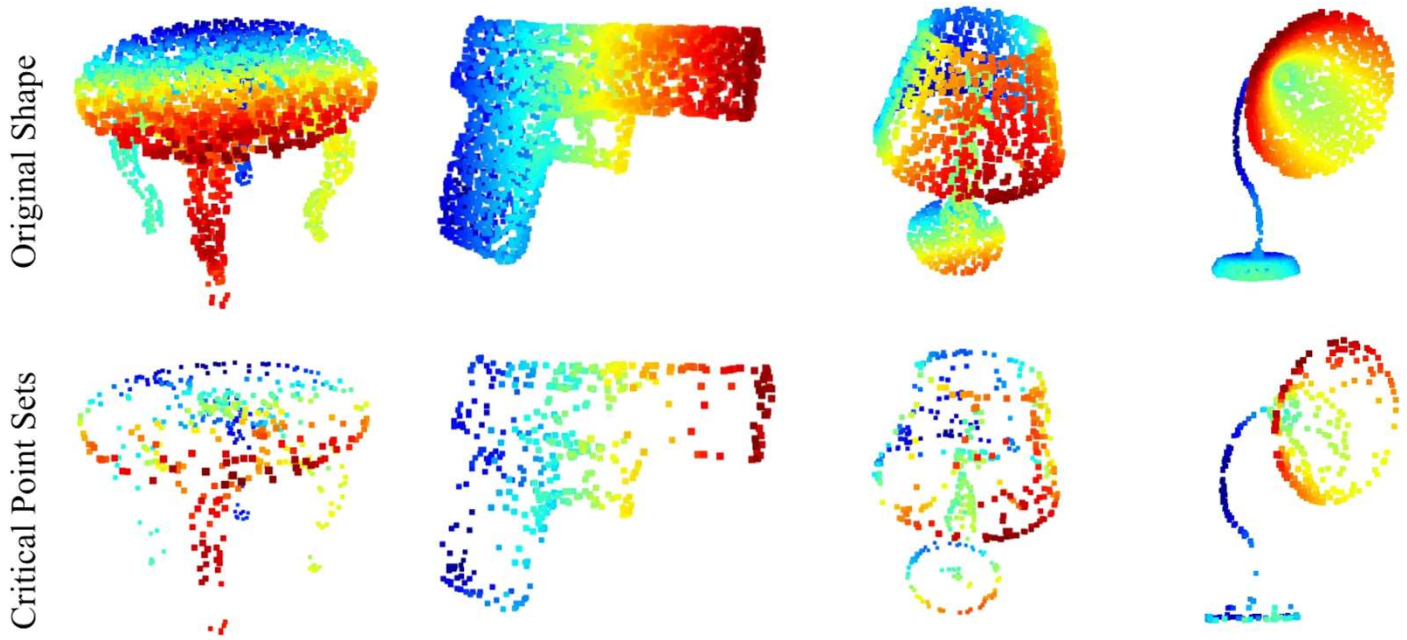
Recall: Voxel Feature Encoder is a mini-PointNet

- VoxelNet – Voxel Feature Encoder



Zhou et al., "**VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection**", *CVPR 2018*

Visualize What is Learned by Reconstruction



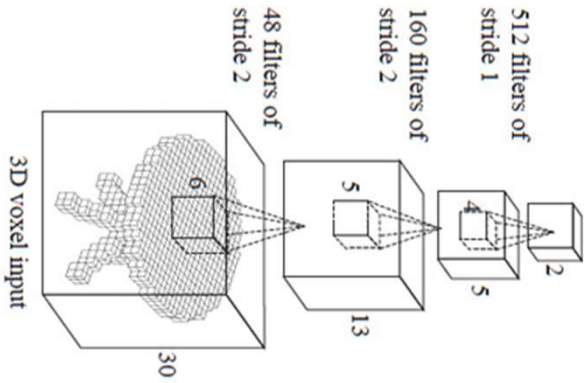
Salient points are discovered!

Limitations of PointNet

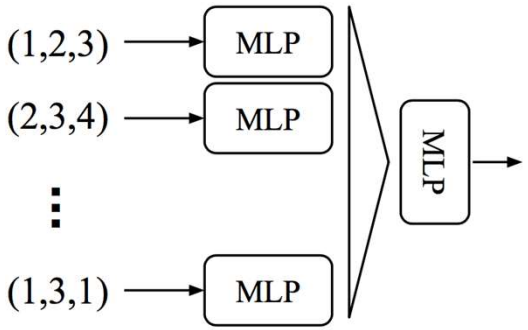
Hierarchical feature learning
Multiple levels of abstraction

v.s.

Global feature learning
Either one point or all points



3D CNN (Wu et al.)

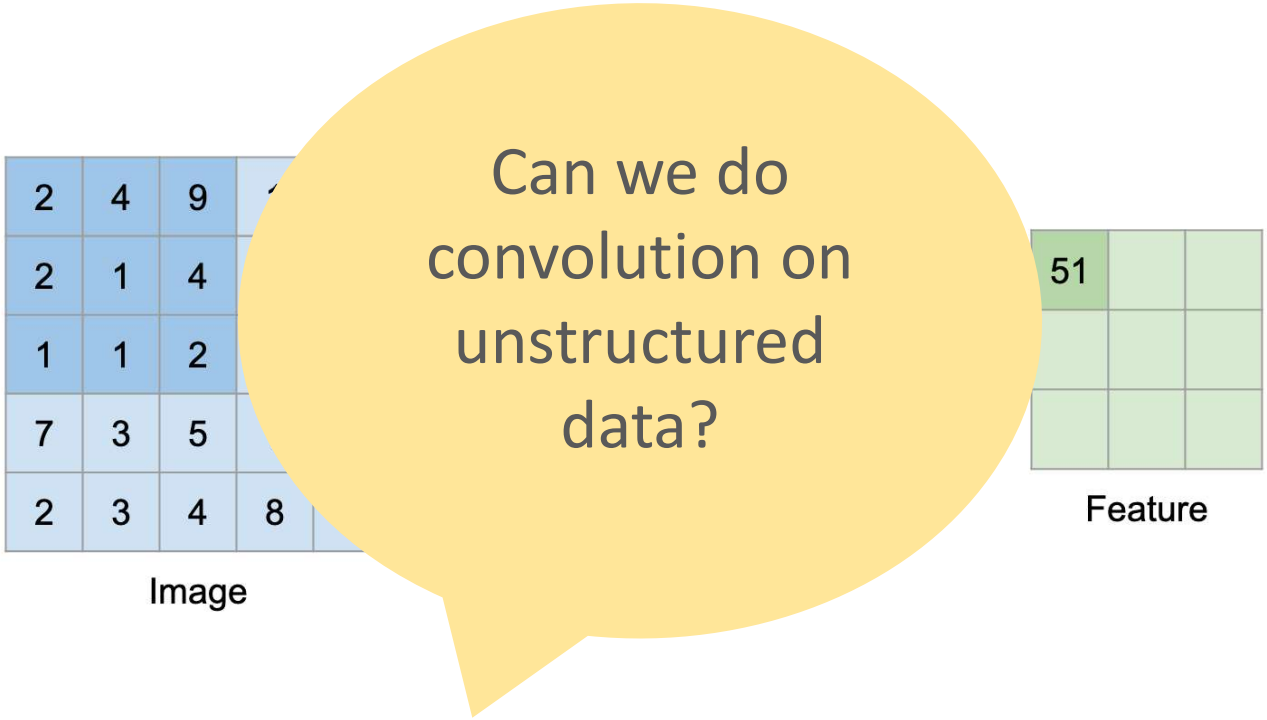


PointNet (vanilla) (Qi et al.)

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

Convolution: aggregating features from local neighborhood

- Find neighbors in metric space – within kxk grid
- Aggregate – weighted-sum

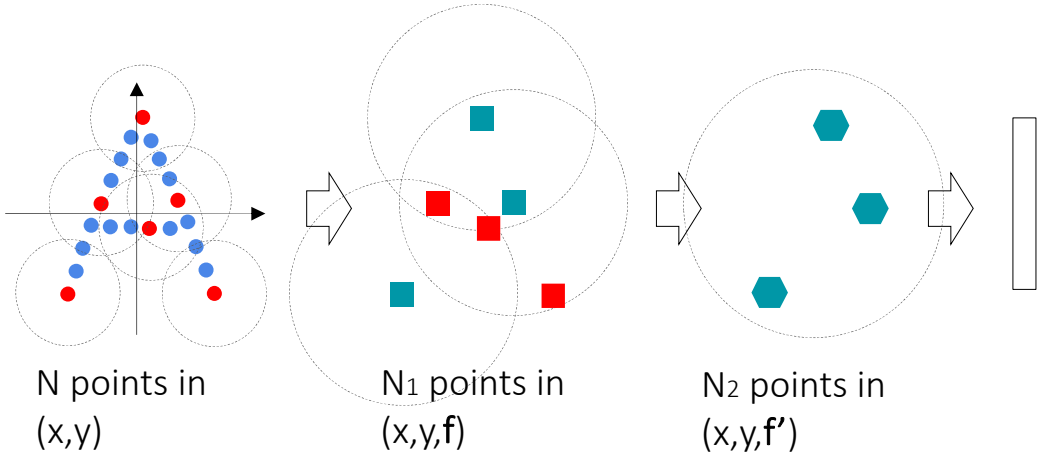


Local Feature Extraction

- A neighboring aggregation scheme (message passing)

  Example neighbors:
  - Regular convolution: 1D/2D/3D spatial grid
  - k-NN query (faster)
  - Ball query (results in more stable features)

PointNet++: Multi-Scale PointNet



Repeat

Sample anchor points

Find neighborhood of anchor points

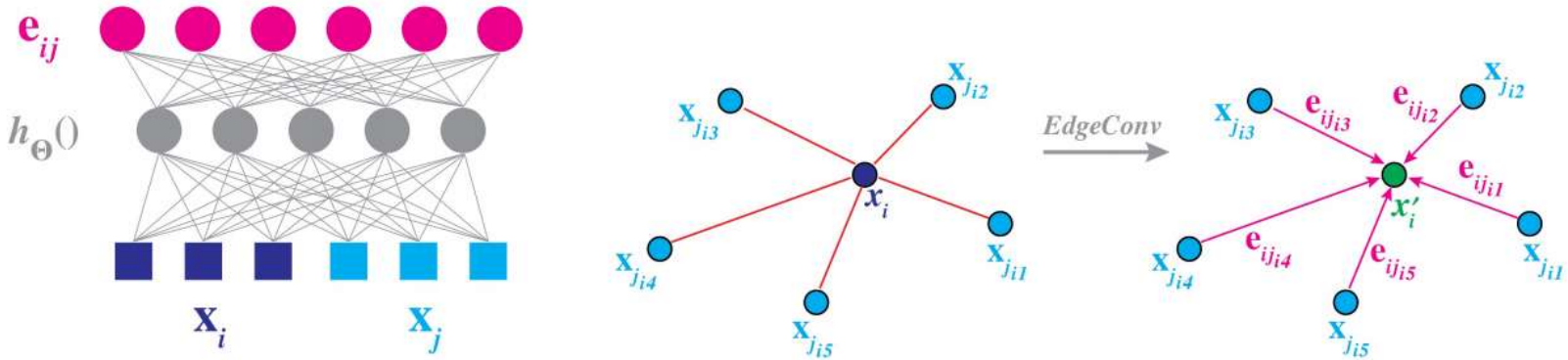
Apply PointNet in each neighborhood to mimic convolution

like grids in convolution kernel

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

Point Convolution As Graph Convolution

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



EdgeConv: $e'_{ijm} = \mathrm{ReLU}(\theta_m \cdot (\mathbf{x}_j - \mathbf{x}_i) + \phi_m \cdot \mathbf{x}_i)$, $x'_{im} = \max_{j:(i,j)\in\mathcal{E}} e'_{ijm}$

PointNet

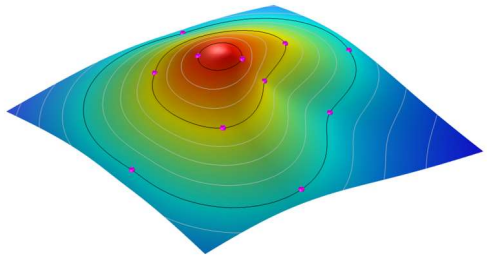
local

global

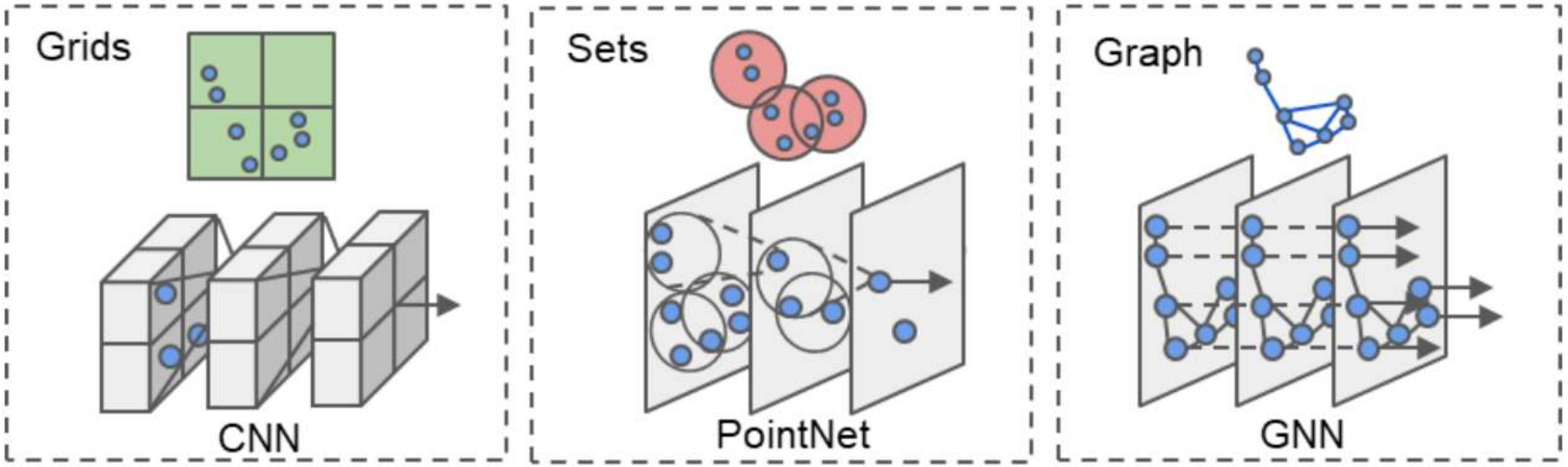
Wang et al., "**Dynamic Graph CNN for Learning on Point Clouds**", *Transactions on Graphics, 2019*
Liu et al., "**Relation-Shape Convolutional Neural Network for Point Cloud Analysis**", *CVPR 2019*
*Shi et al., "***Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud***", CVPR 2020

Regular Convolution, Point Convolution are special cases of Spatial Graph Convolution

	Regular Conv	Point Conv	Spatial Graph Conv
How to construct the graph (define neighborhood)	kxk spatial grid, w/o edge	range query, w/o edge	range query, w/ edge
How to design kernel (aggregate)	Learnable Linear Combination with no constraints	Learnable Linear Combination with no constraints	Usually Radial Basis Functions that considers node interactions (edges)



Comparison on Performance



- ✓ Efficient
- ✓ Scalable
- ✓ Feature Hierarchy
- ✗ Needs voxelization
- ✗ Trade-off between voxel resolution and memory
- ✗ Poor performance on objects smaller than a voxel

- ✓ Feature Hierarchy with set abstraction
- ✗ Slow (sampling)
- ✗ Receptive field is large
- ✗ Poor performance on small objects (ignored during sampling)

- ✓ Feature Hierarchy
- ✗ Slow (sampling)
- ✗ Poor performance on small objects (ignored during sampling)

Tasks: Reconstruction, Shape Completion/Manipulation

Inputs

Point Clouds or
Meshes or
Volumetric Grids or
2D Images

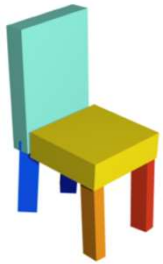
Generative Models

GAN or
(Graph) VAE

Latent/Output Representation

$$F(x) = 0$$

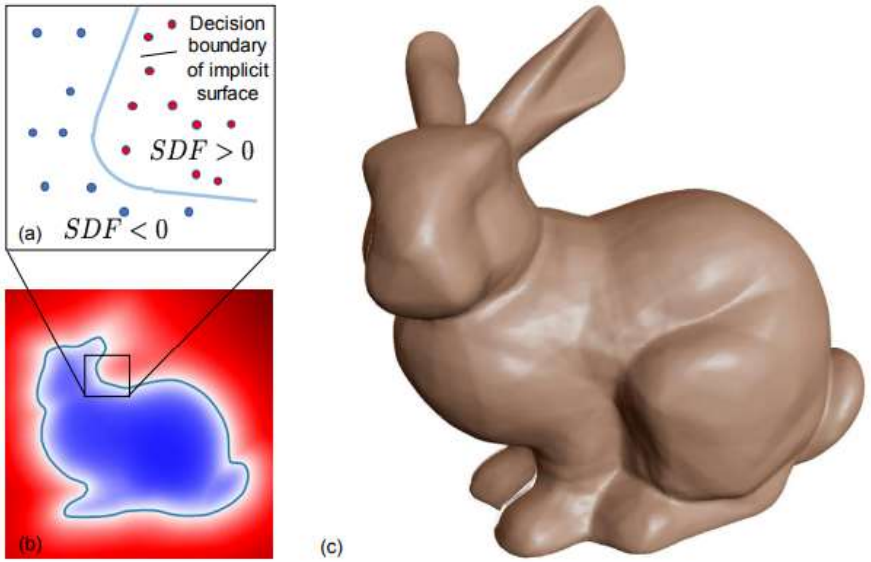
Implicit Shape or

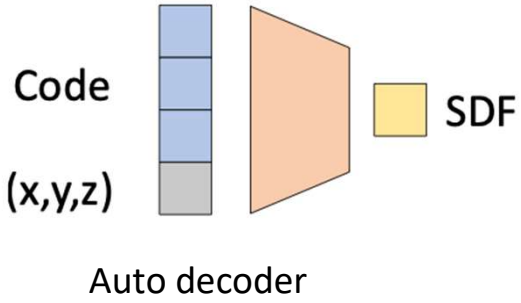


Part Assembly

Implicit Surface Reconstruction

- Implicit field function $F(x)$ (e.g., signed distance)
- Extract the iso-surface $F(x) = 0$





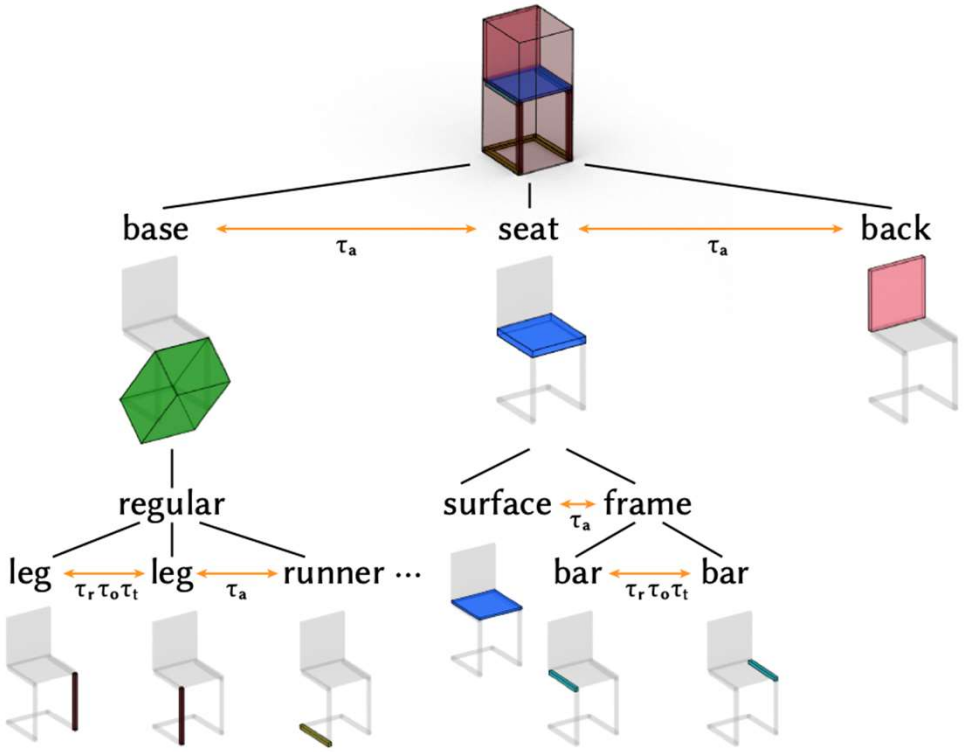
Park et al., "**DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation**", *CVPR 2019*
Other two similar paper on implicit representation:
Mescheder et al., "**Occupancy Networks: Learning 3D Reconstruction in Function Space**", *CVPR 2019*
Chen et al., "**Learning Implicit Fields for Generative Shape Modeling**", *CVPR 2019*

- In general,
  - First map the input to a shape embedding
  - Then reconstruct by decoding

- Limitation
  - Output is not explicitly grounded on the input
  - Structures of 3D objects are not explicitly leveraged
  - Cannot generalize to unseen objects

Structured Prediction: Part-based

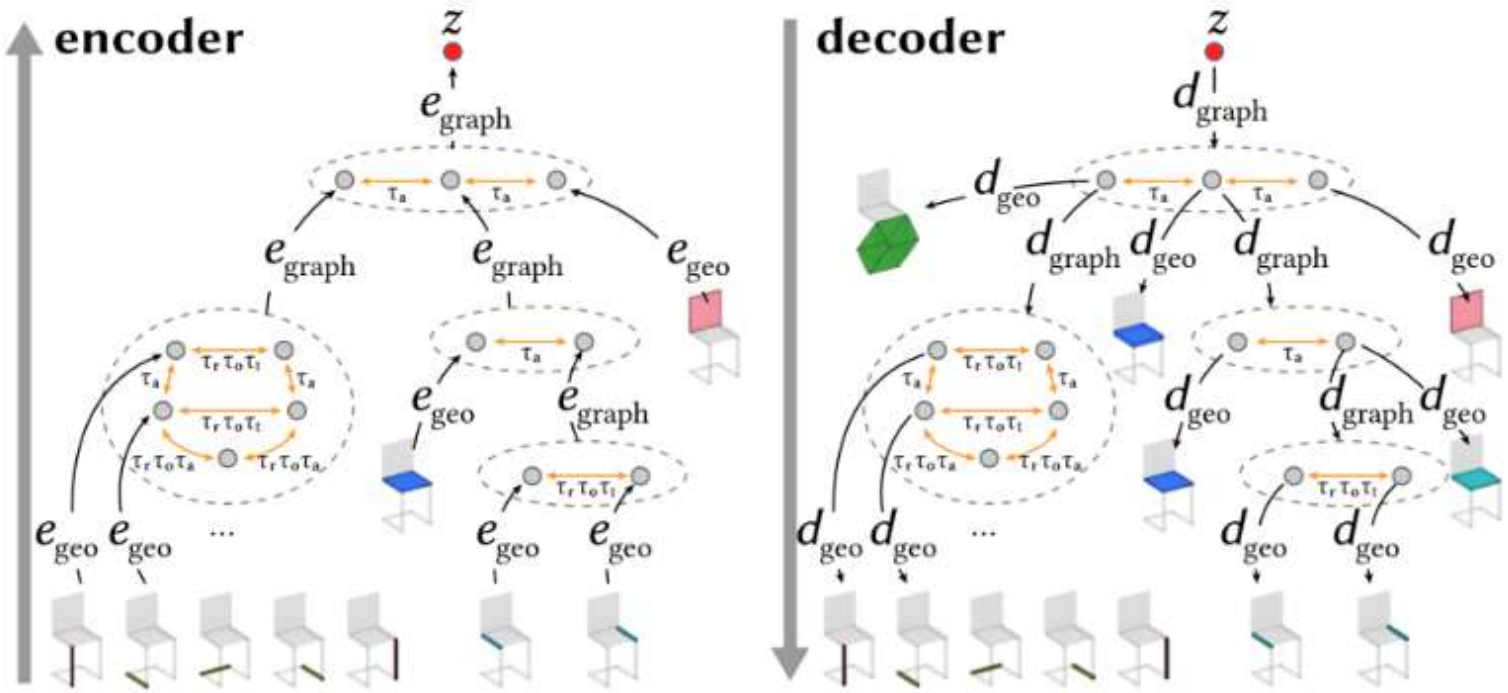
Hierarchical Graph



Mo et al., "**StructureNet, a hierarchical graph network for learning PartNet shape generation**", *Siggraph Asia 2019*

Structured Prediction: Part-based

Recursive Network for Hierarchical Graph AE



Mo et al., "**StructureNet, a hierarchical graph network for learning PartNet shape generation**", *Siggraph Asia 2019*

Structured Prediction: Part-based

Mo et al., "**StructureNet, a hierarchical graph network for learning PartNet shape generation**", *Siggraph Asia 2019*

Summary

- Six 3D Data Representations
- Learning from Irregular Data:
  1) Converting to Regular Data (Volumetric Grids) + Grid CNN or
  2) Directly Learning from Sets(Points) + PointNet or
  3) Directly Learning from Constructed Graphs + Graph CNN

Or combinations of 1), 2) and 3)