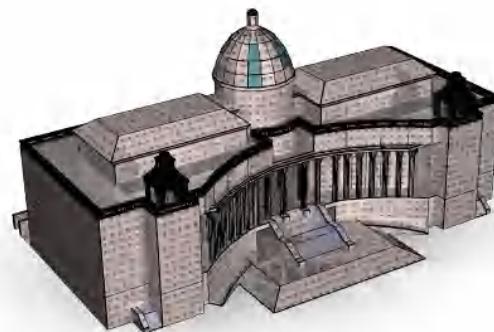
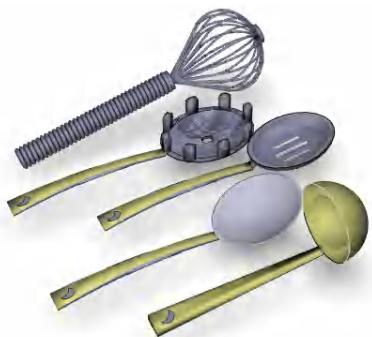
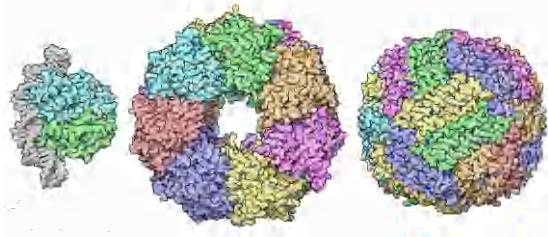




2019 Tutorial on 3D Deep Learning

Hao Su (UCSD)



Broad Applications of 3D data



Robotics



Broad Applications of 3D data



Robotics



Augmented
Reality



Broad Applications of 3D data



Robotics



Augmented
Reality



Autonomous
driving

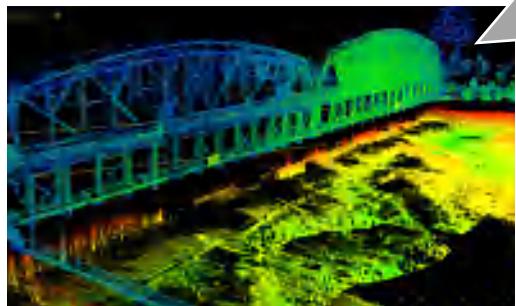
Broad Applications of 3D data



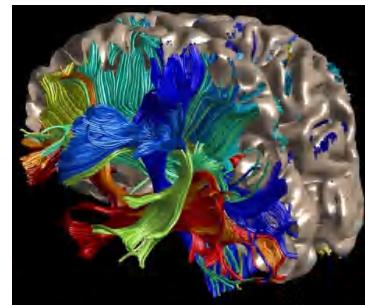
Robotics



Augmented Reality



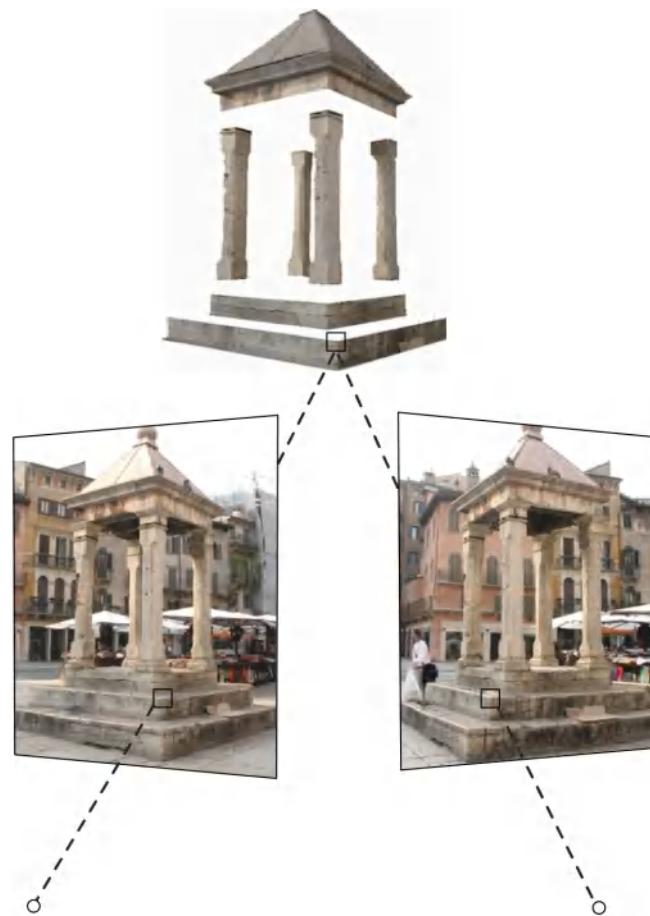
Autonomous driving



Medical Image Processing

Traditional 3D Vision

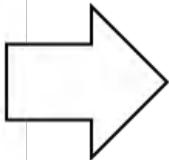
Multi-view Geometry: Physics based



3D Learning: Knowledge Based

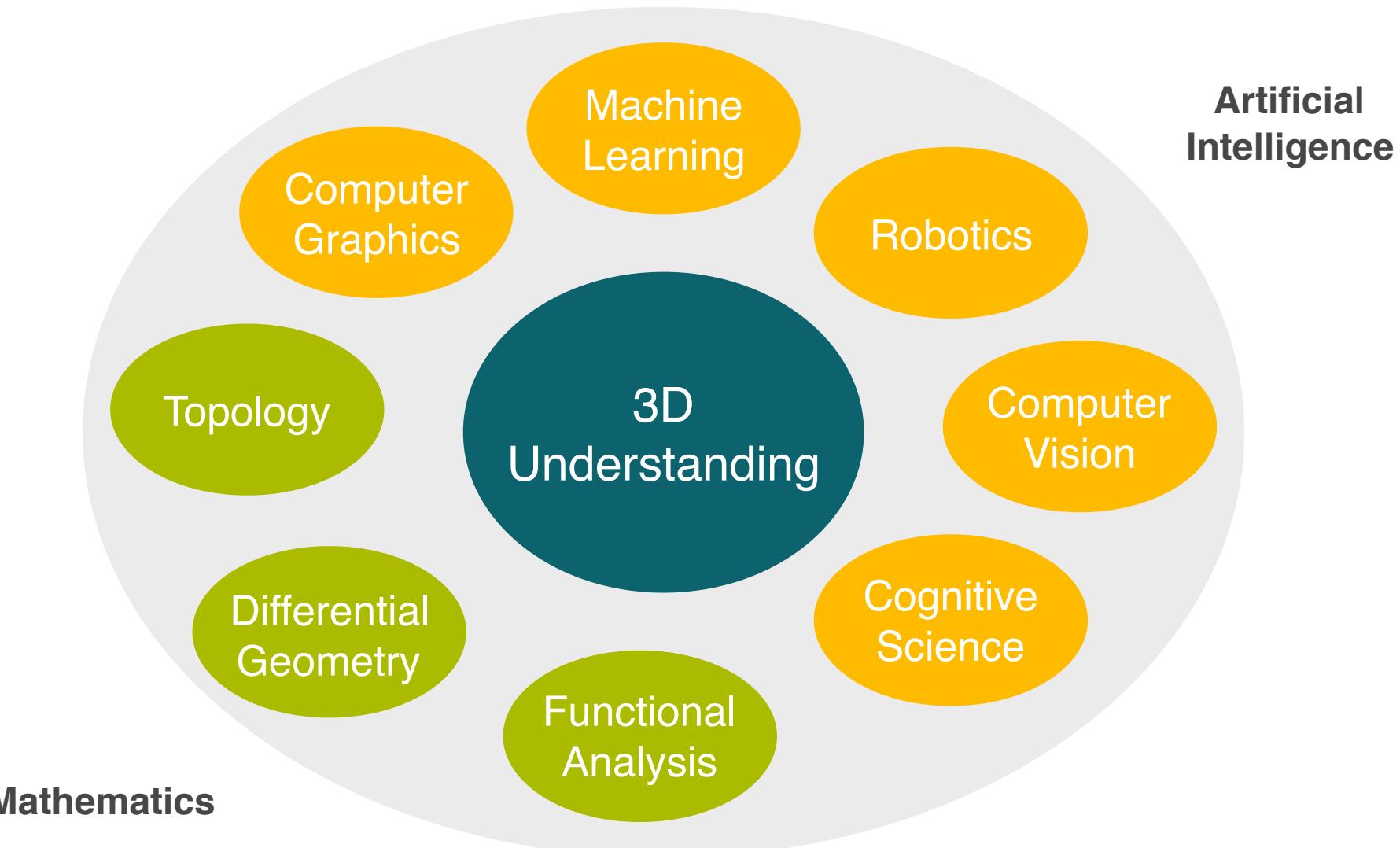


Acquire Knowledge of 3D World by Learning



A priori knowledge of
the 3D world

A New Rising Field

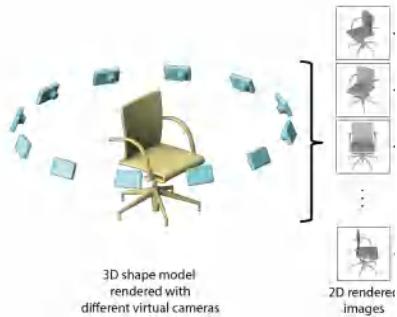


The Representation Challenge of 3D Deep Learning

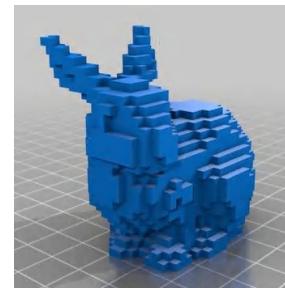
Rasterized form
(regular grids)

Geometric form
(irregular)

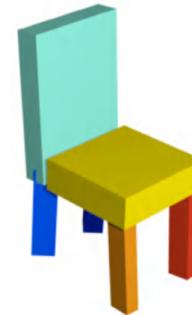
The Representation Challenge of 3D Deep Learning



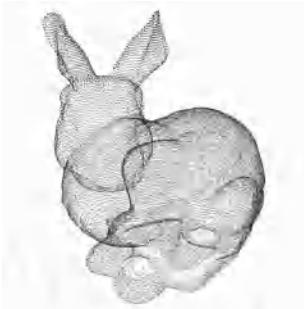
Multi-view



Volumetric



Part Assembly



Point Cloud



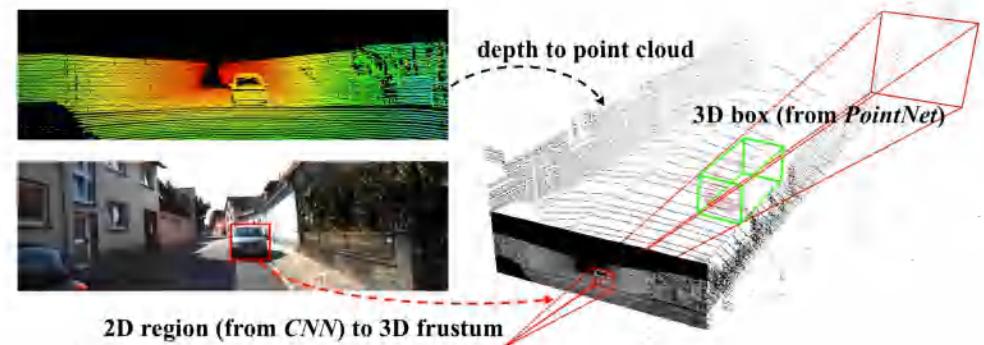
Mesh (Graph CNN)

$$F(x) = 0$$

Implicit Shape

The Richness of 3D Learning Tasks

3D Analysis



Detection



Classification



Segmentation (object/scene)



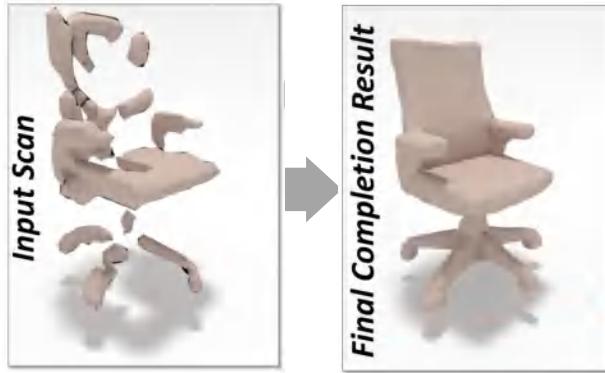
Correspondence

The Richness of 3D Learning Tasks

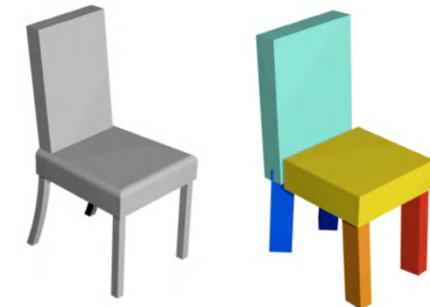
3D Synthesis



Monocular
3D reconstruction



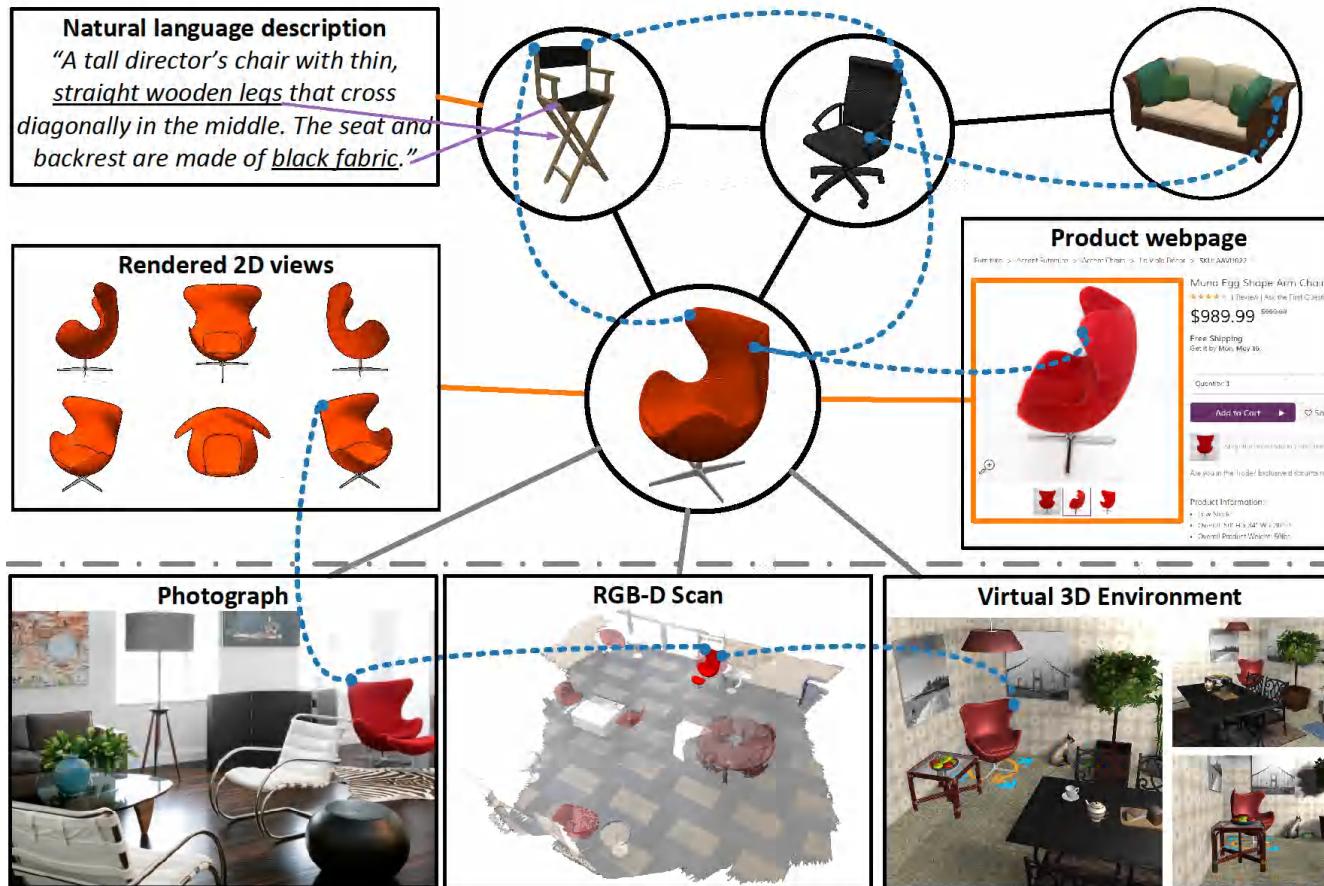
Shape completion



Shape modeling

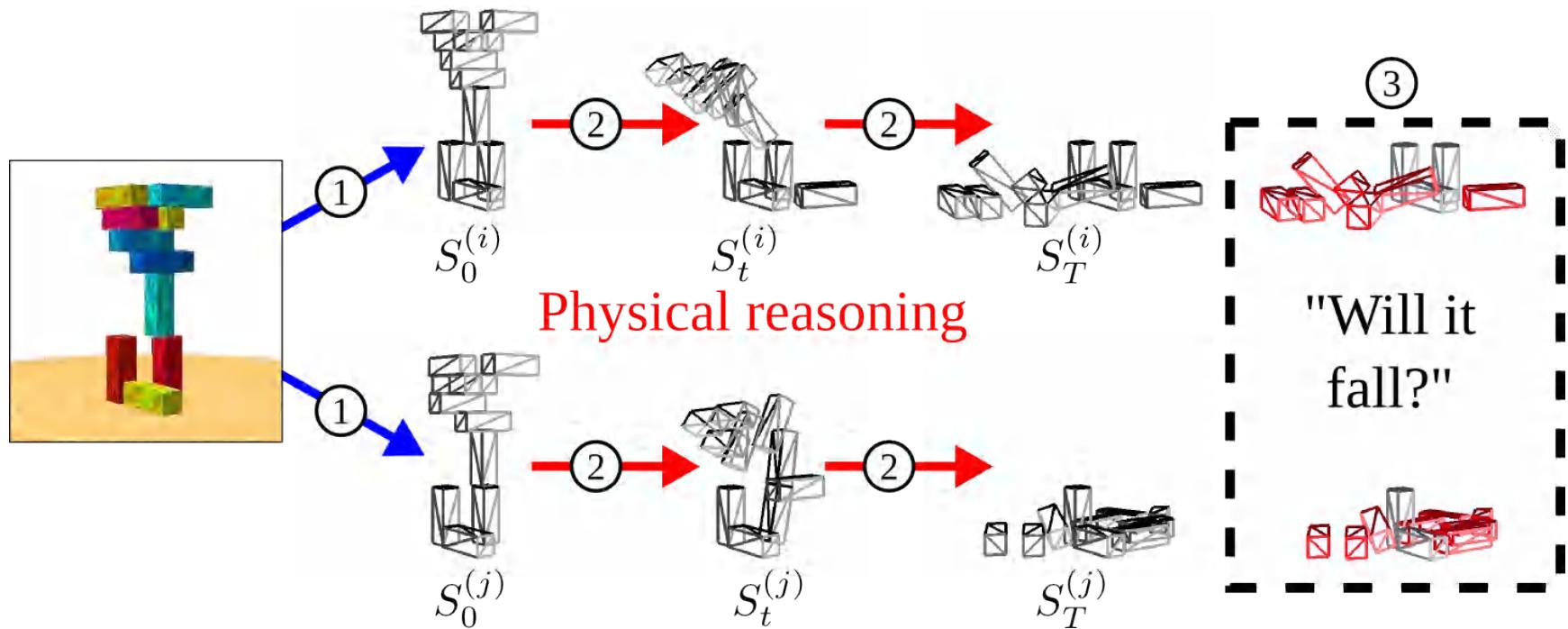
The Richness of 3D Learning Tasks

3D-based Knowledge Transportation

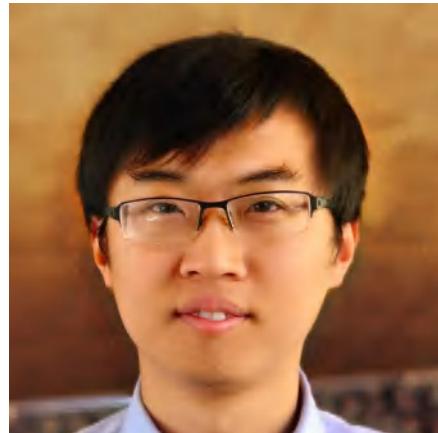


3D Learning Tasks

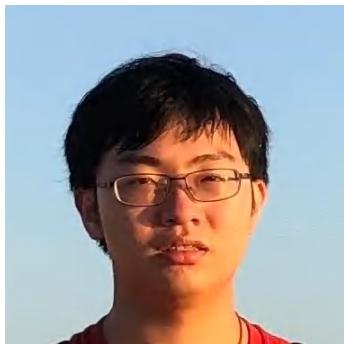
From static to dynamic



Speaker



Team Members



Shuo Cheng



Siyu Hu



Zetian Jiang

Algorithms of 3D Deep Learning

Hao Su (UCSD)

Topics

- Classification
- Segmentation and Detection
- Reconstruction
- 3D Dataset
- 3D Few-shot Learning

Topics

- Classification
- Segmentation and Detection
- Reconstruction
- 3D Dataset
- 3D Few-shot Learning

Task: 3D Classification



This is a chair!

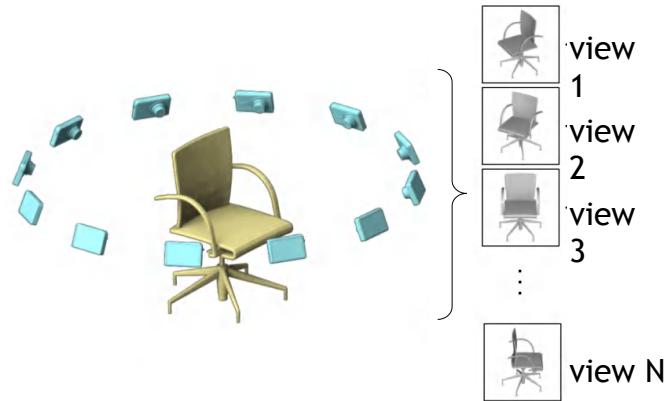
Covered methods: Volumetric CNN, OctNet, O-CNN, SparseConvNet, PointNet, PointNet++, RS CNN, DGCNN, Point ConvNet, KPConv, Monte Carlo Point Convolution, PConv, Multi-View CNN, Spectral CNN, Synchronized Spectral CNN, Spherical CNN

Multi-View CNN

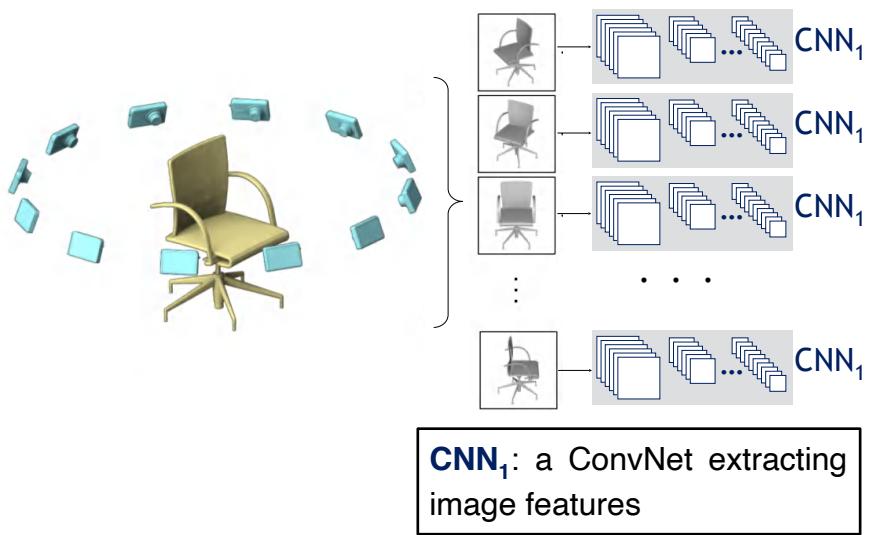
Given an Input Shape



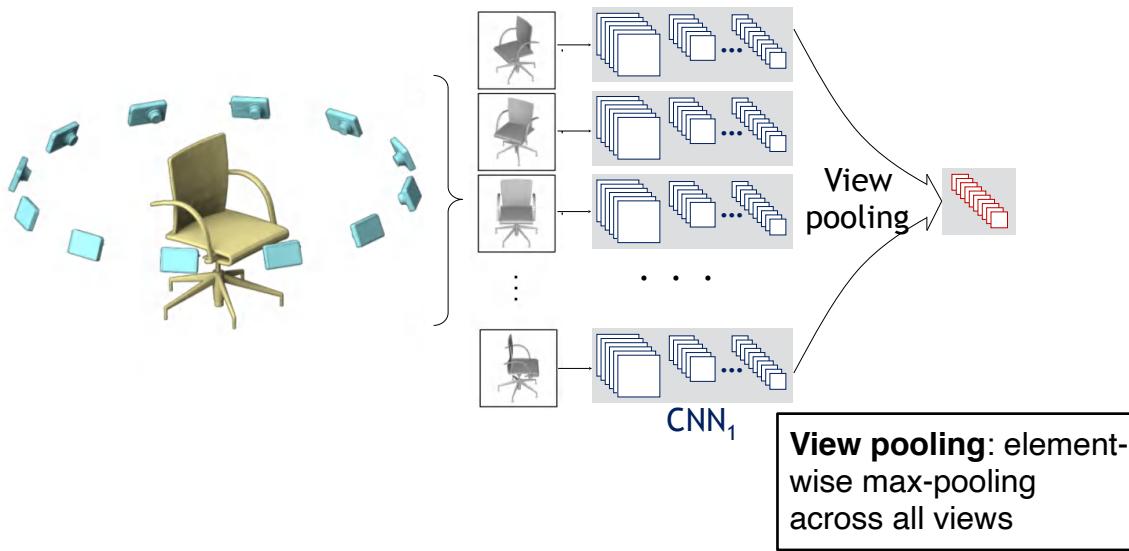
Render with Multiple Virtual Cameras



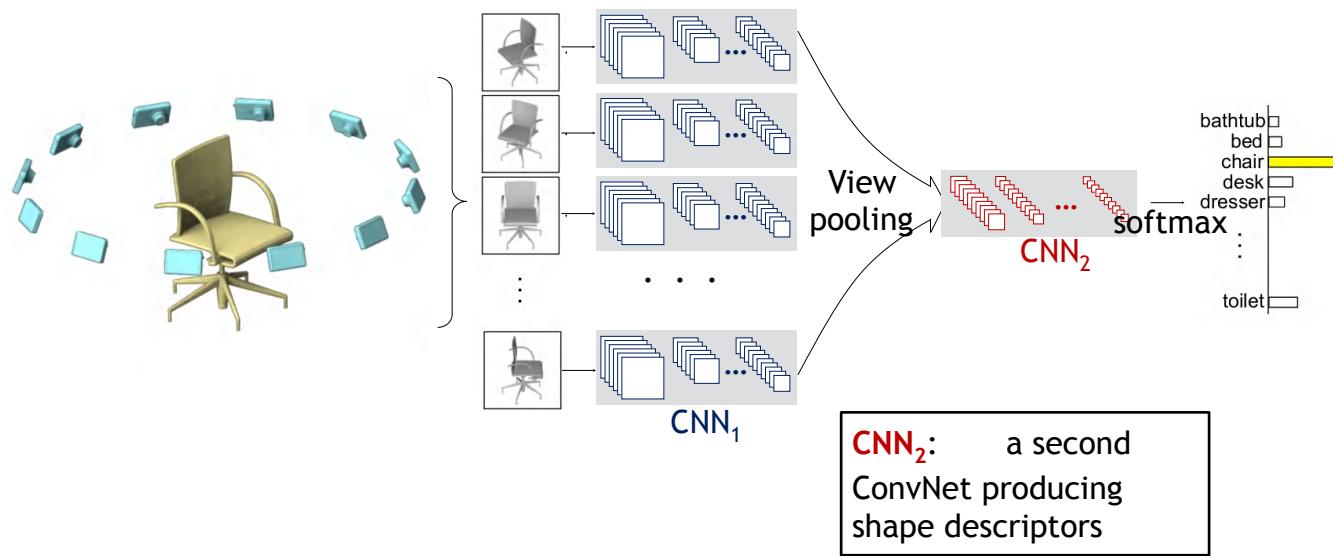
The Rendered Images are Passed through CNN_1 for Image Features



All Image Features are Combined by View Pooling



... and then Passed through CNN_2 and to Generate Final Predictions



Experiments – Classification & Retrieval

Method	Classification (Accuracy)	Retrieval (mAP)
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

[credit: Hang Su]

- Indeed gives good performance
- Can leverage vast literature of image classification
- Can use pertained features
- Need projection
- What if the input is noisy and/or incomplete? e.g., point cloud

Volumetric CNN

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

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

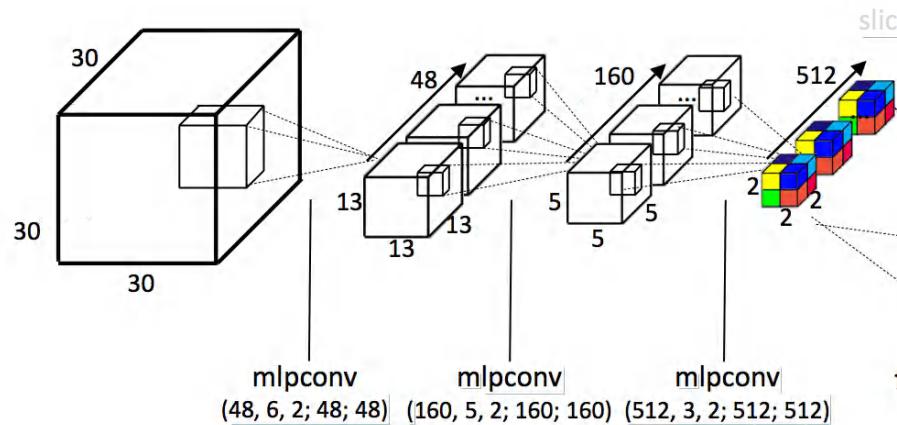
Voxelization

Represent the occupancy of regular 3D grids

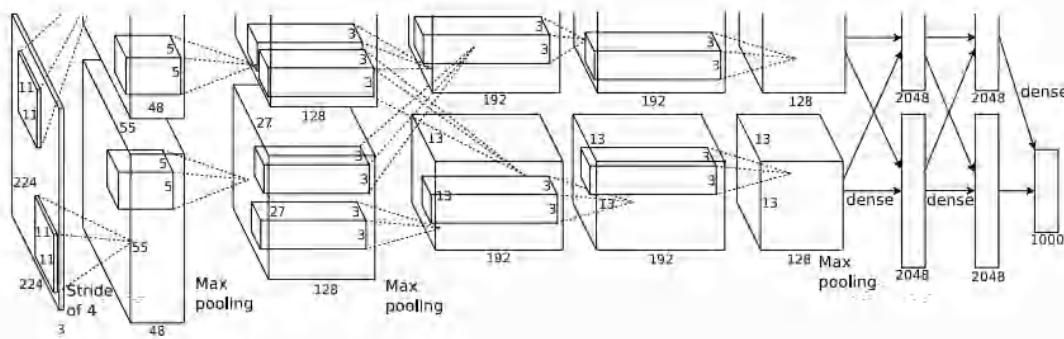


3D CNN on Volumetric Data

3D convolution uses 4D kernels



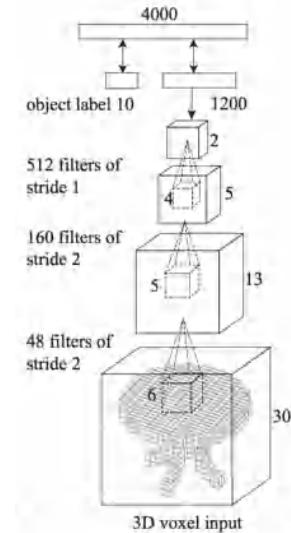
Complexity Issue



AlexNet, 2012

Input resolution: 224x224

$$224 \times 224 = 50176$$

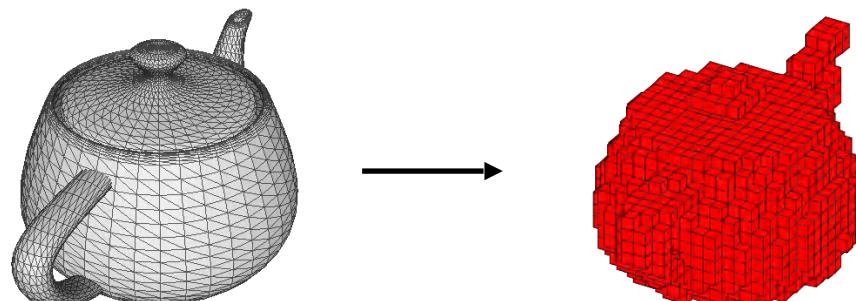


3DShapeNets,
2015

Input resolution: 30x30x30

$$224 \times 224 = 27000$$

Complexity Issue



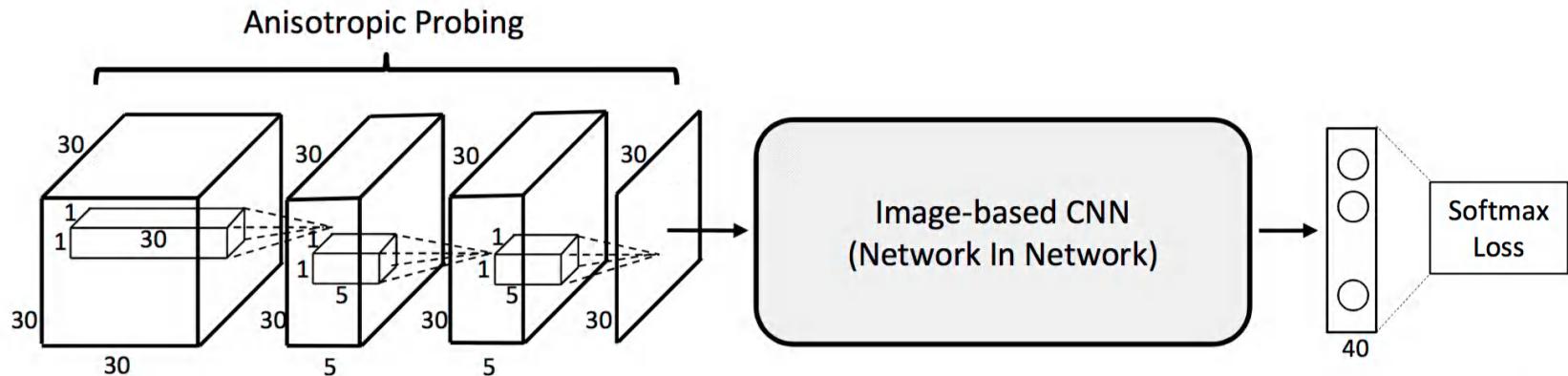
Polygon Mesh

Occupancy Grid
 $30 \times 30 \times 30$

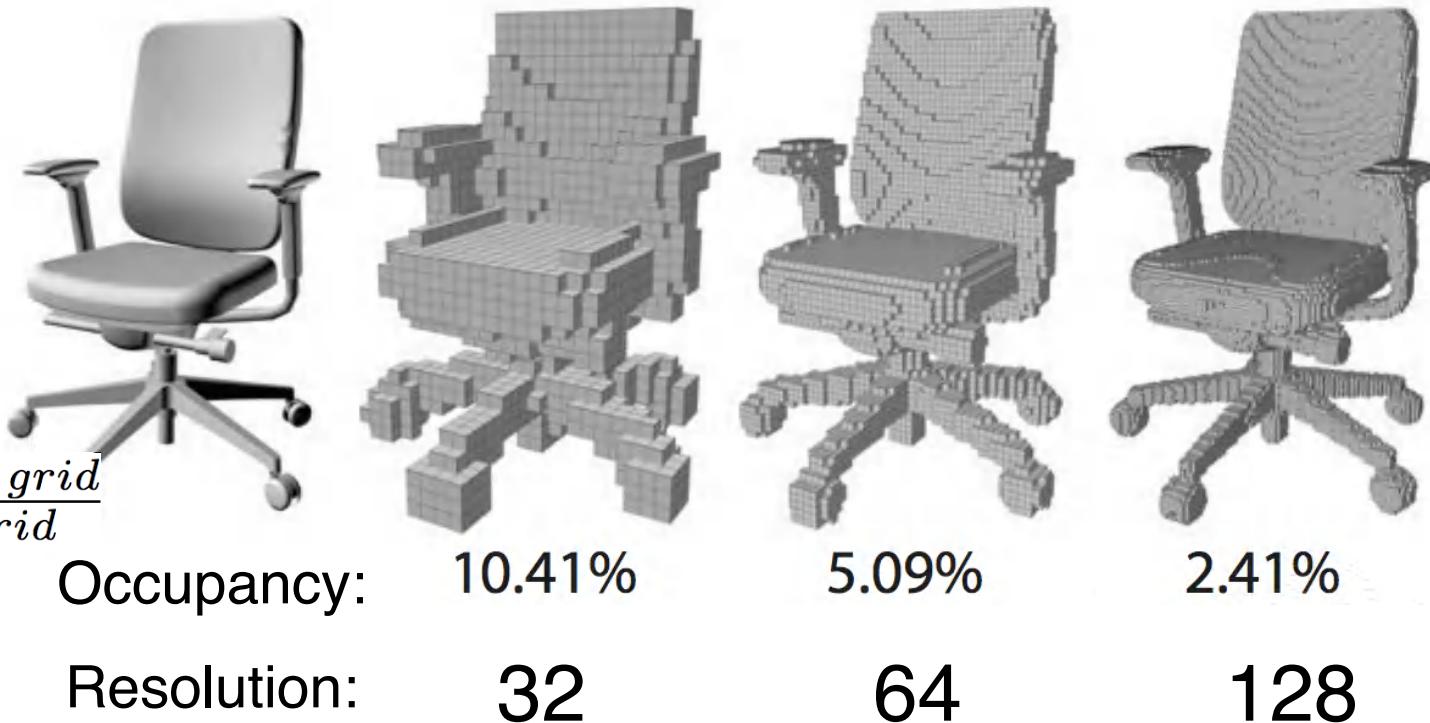
Information loss in voxelization

Idea 1: Anisotropic Probing

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

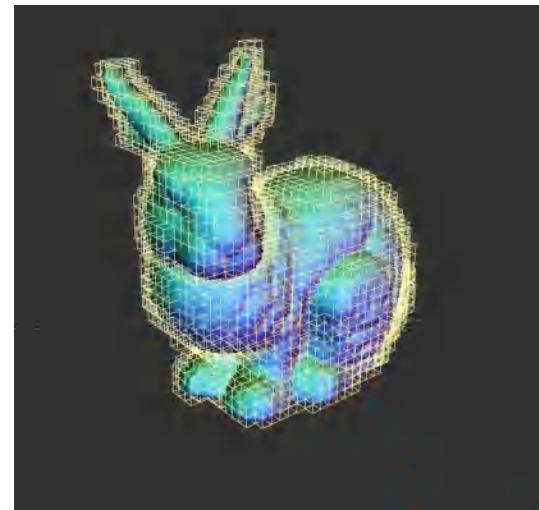
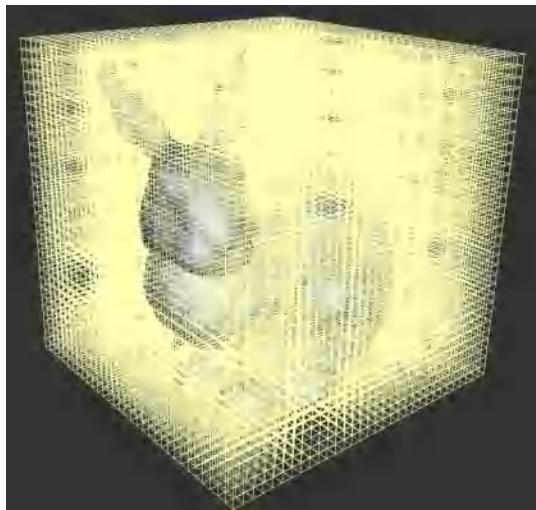


More Principled: Sparsity of 3D Shapes



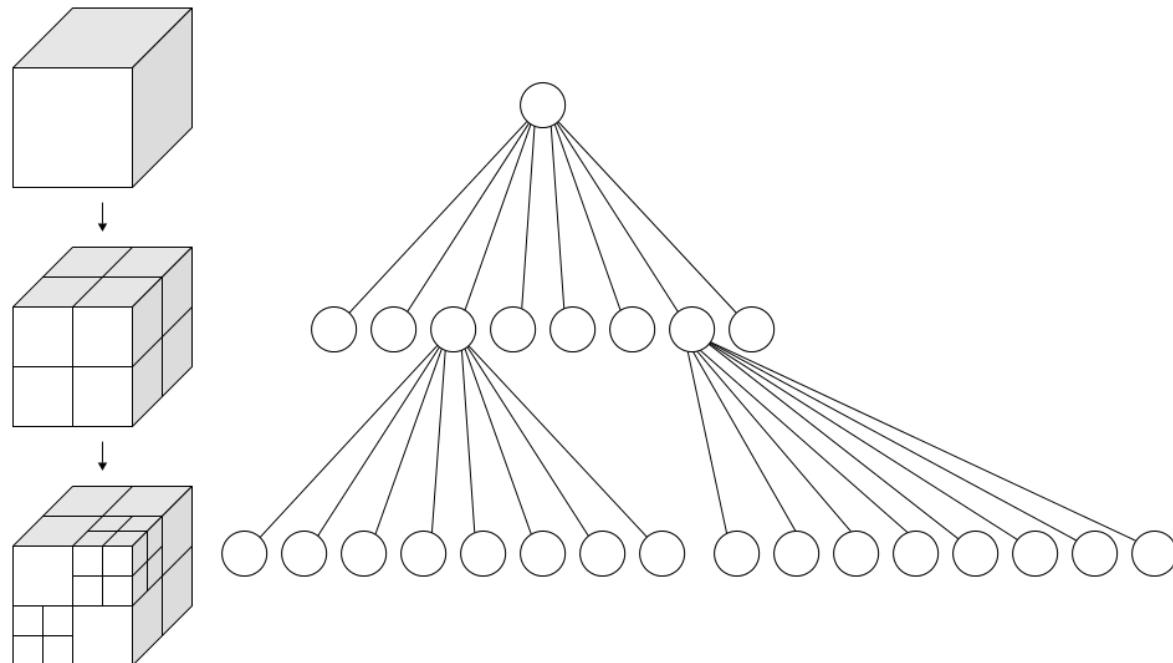
Store only the Occupied Grids

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



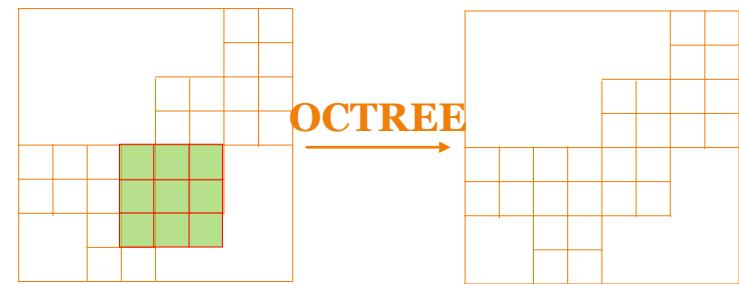
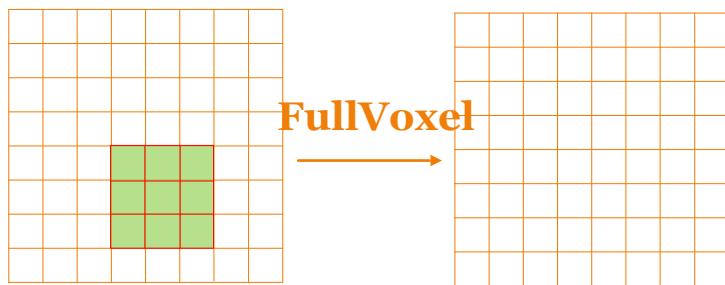
Octree: Recursively Partition the Space

Each **internal node** has exactly eight **children**

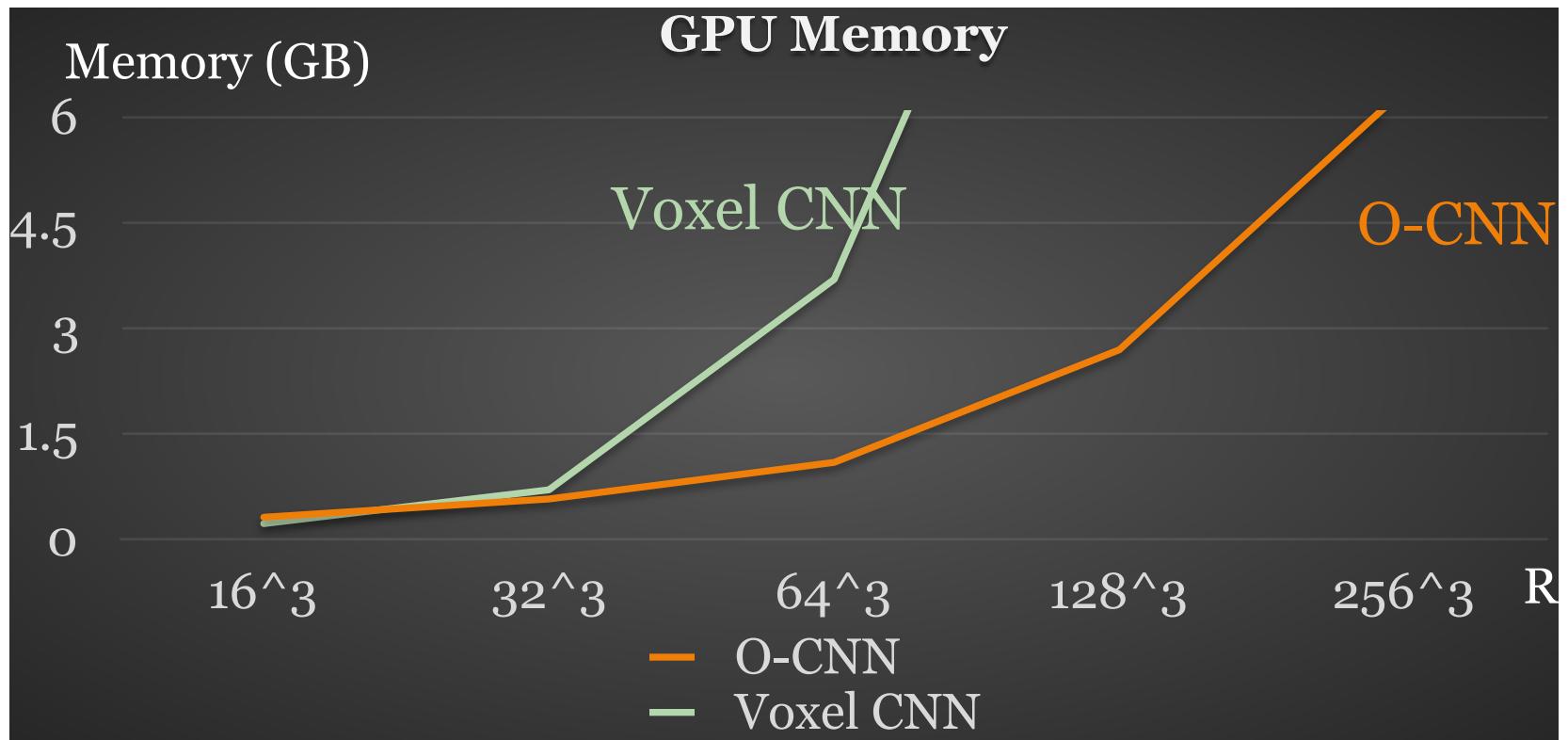


Convolution on Octree

Neighborhood searching: Hash table



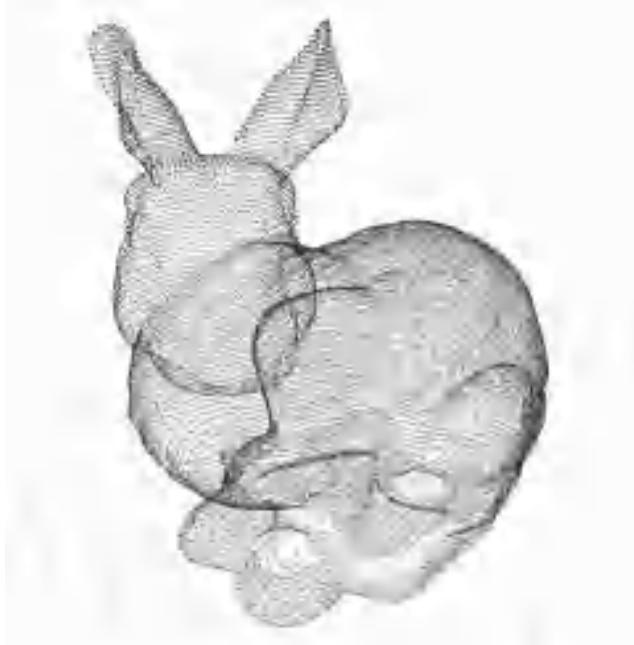
Memory Efficiency



Implementation

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

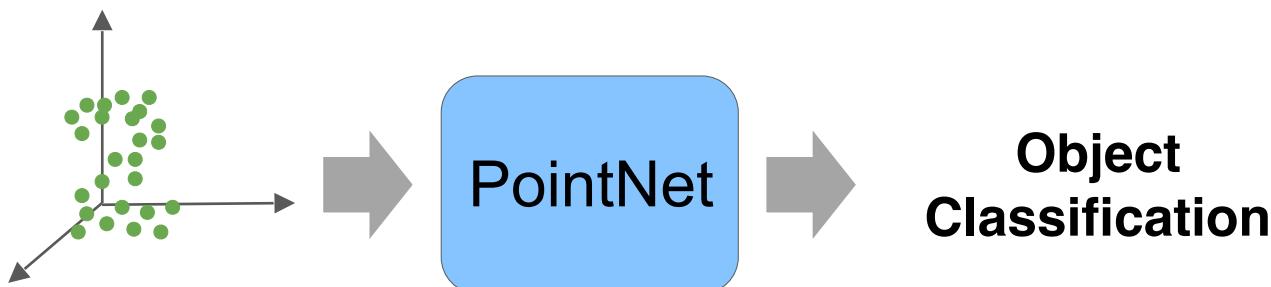
Point Networks



Point cloud
(The most common 3D sensor data)

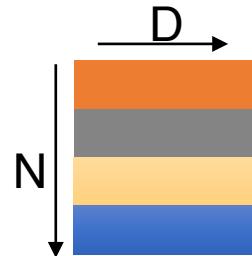
Directly Process Point Cloud Data

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



Permutation invariance

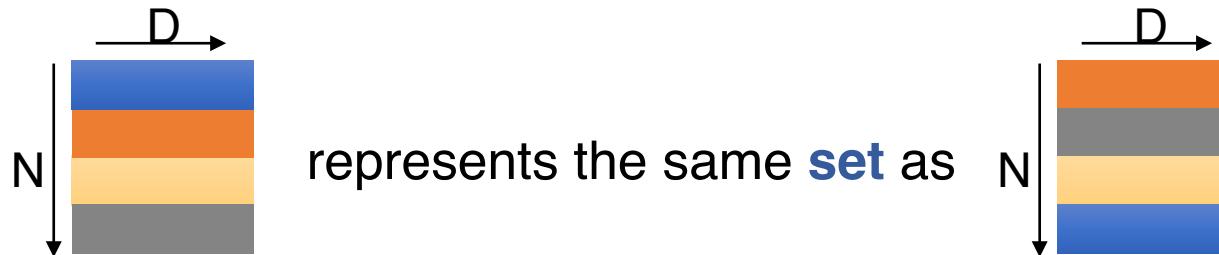
Point cloud: N **orderless** points, each represented by a D dim coordinate



2D array representation

Permutation invariance

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



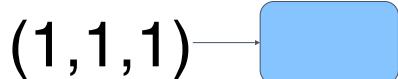
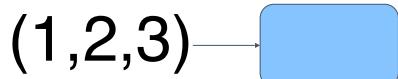
2D array representation

Construct a Symmetric Function

Observe:

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

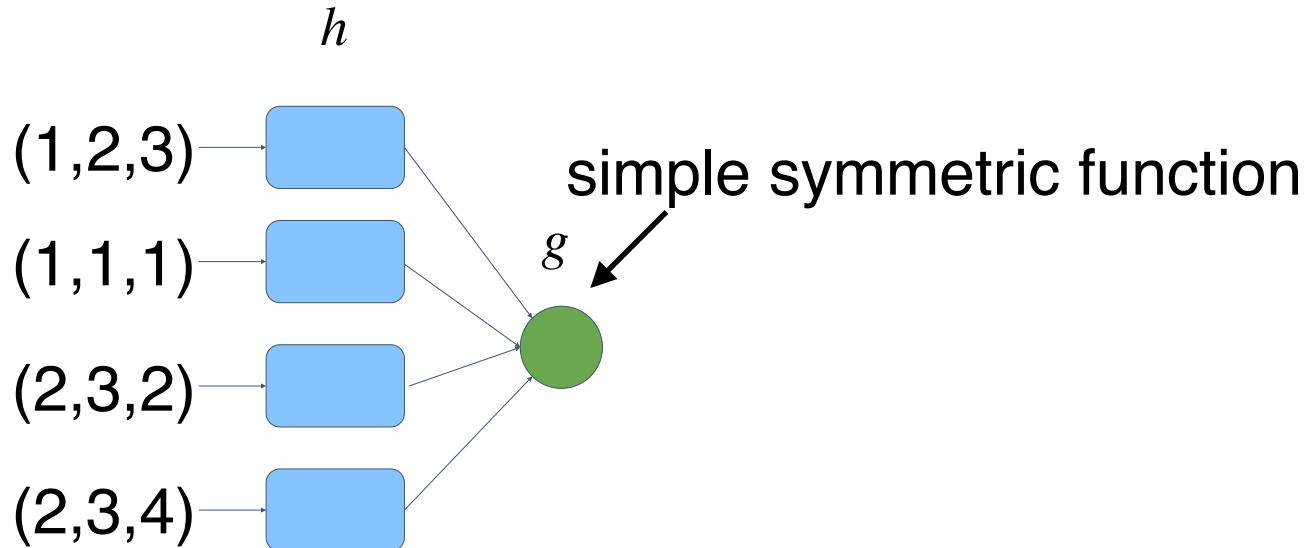
h



Construct a Symmetric Function

Observe:

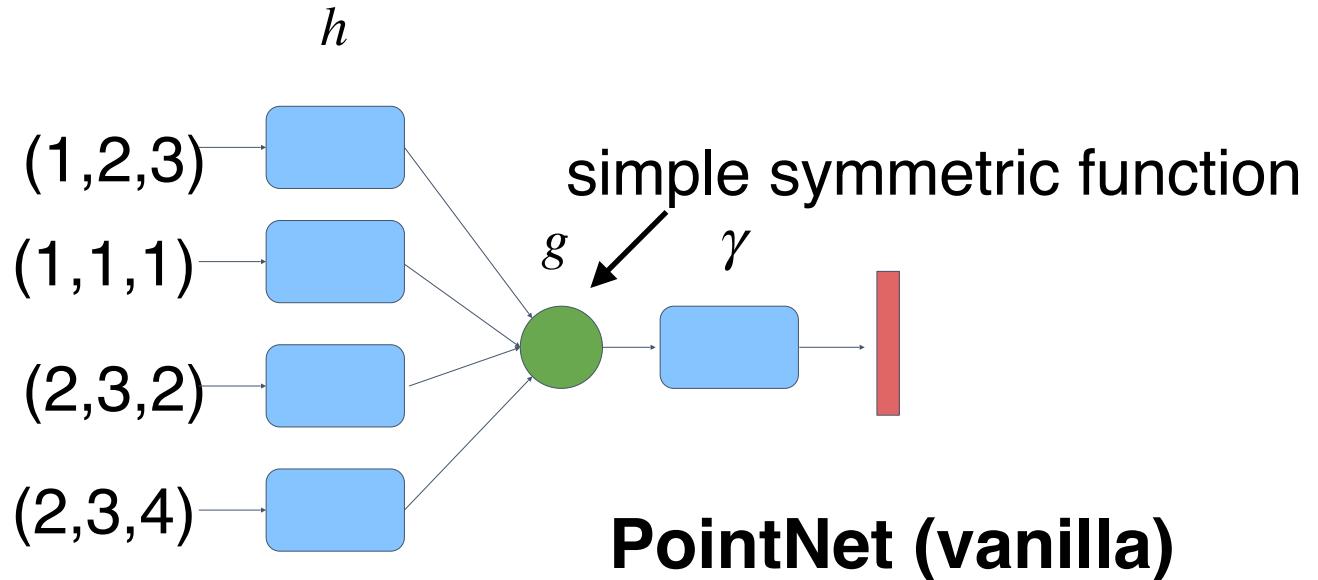
$f(x_1, x_2, \dots, x_n) = \gamma \circ g(h(x_1), \dots, h(x_n))$ is symmetric if g is symmetric



Construct a Symmetric Function

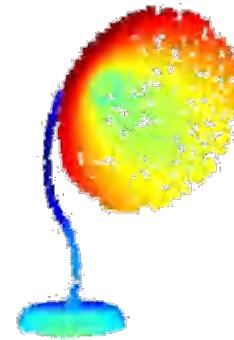
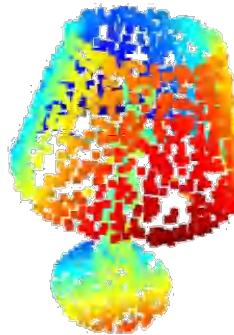
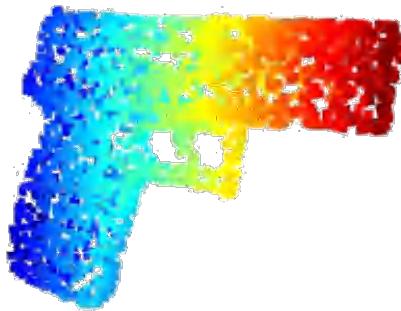
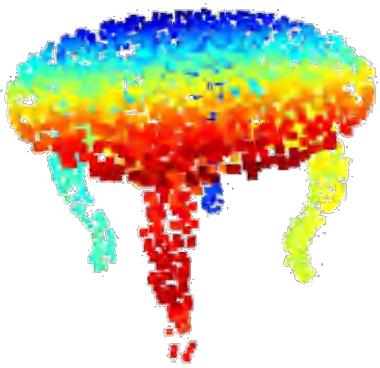
Observe:

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

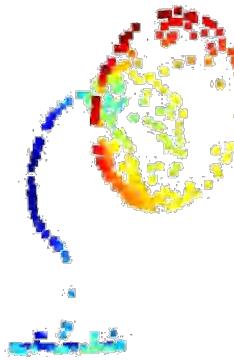
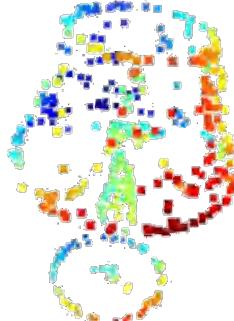
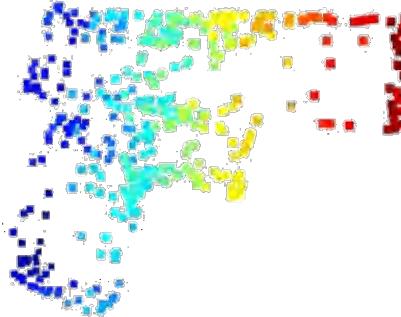
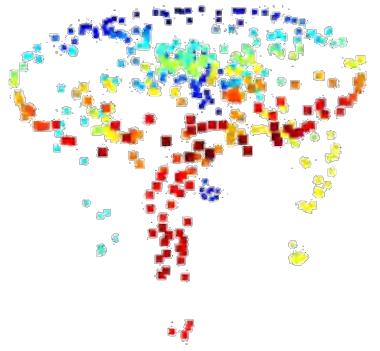


Visualize What is Learned by Reconstruction

Original Shape



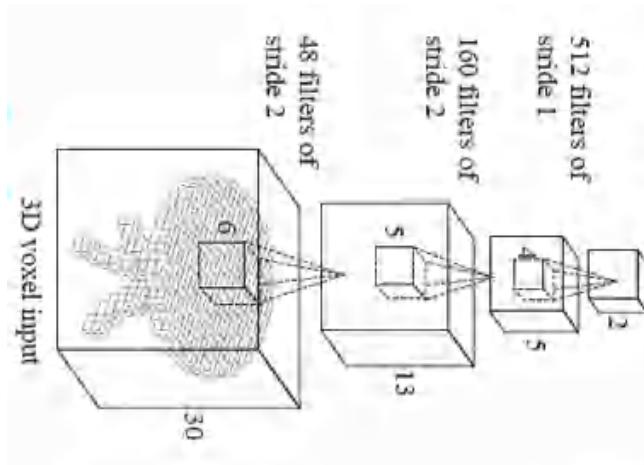
Critical Point Sets



Salient points are discovered!

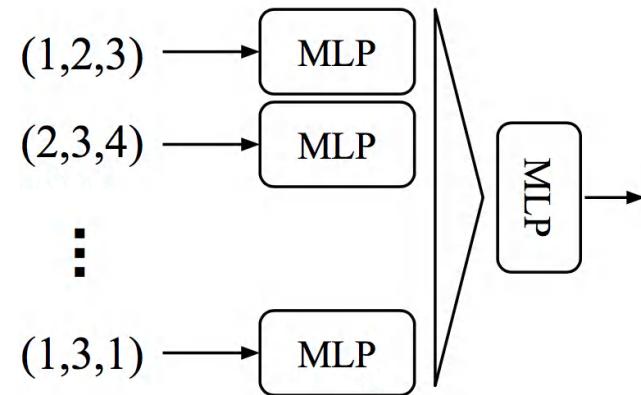
Limitations of PointNet

Hierarchical feature learning
Multiple levels of abstraction



3D CNN (Wu et al.)

Global feature learning
Either one point or all points



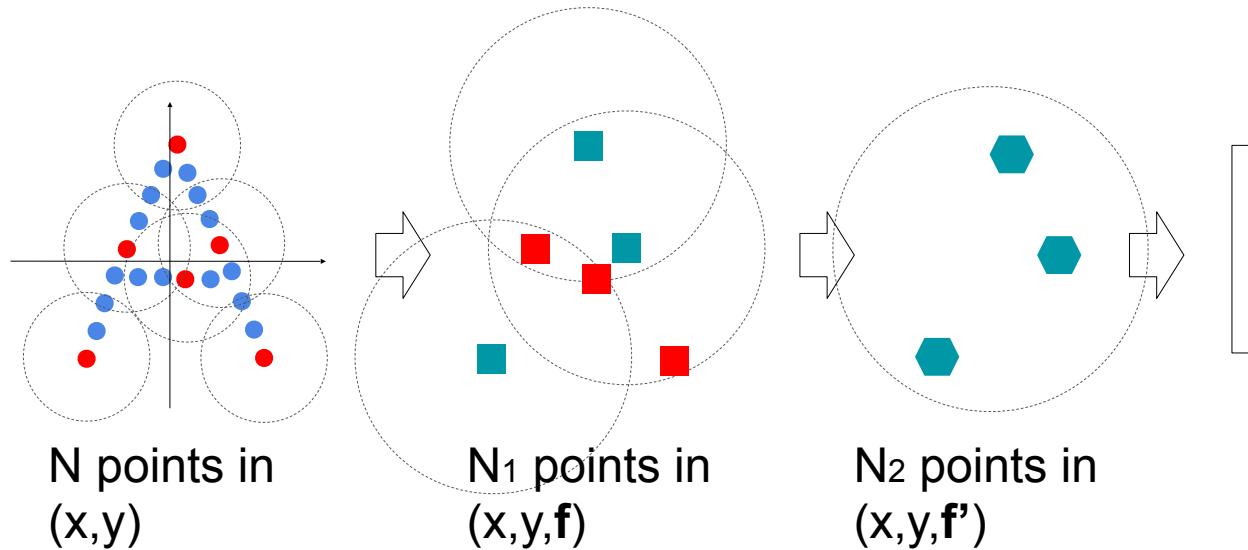
PointNet (vanilla) (Qi et al.)

- 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 generally)
 - k-NN query (faster)

PointNet v2.0: Multi-Scale PointNet

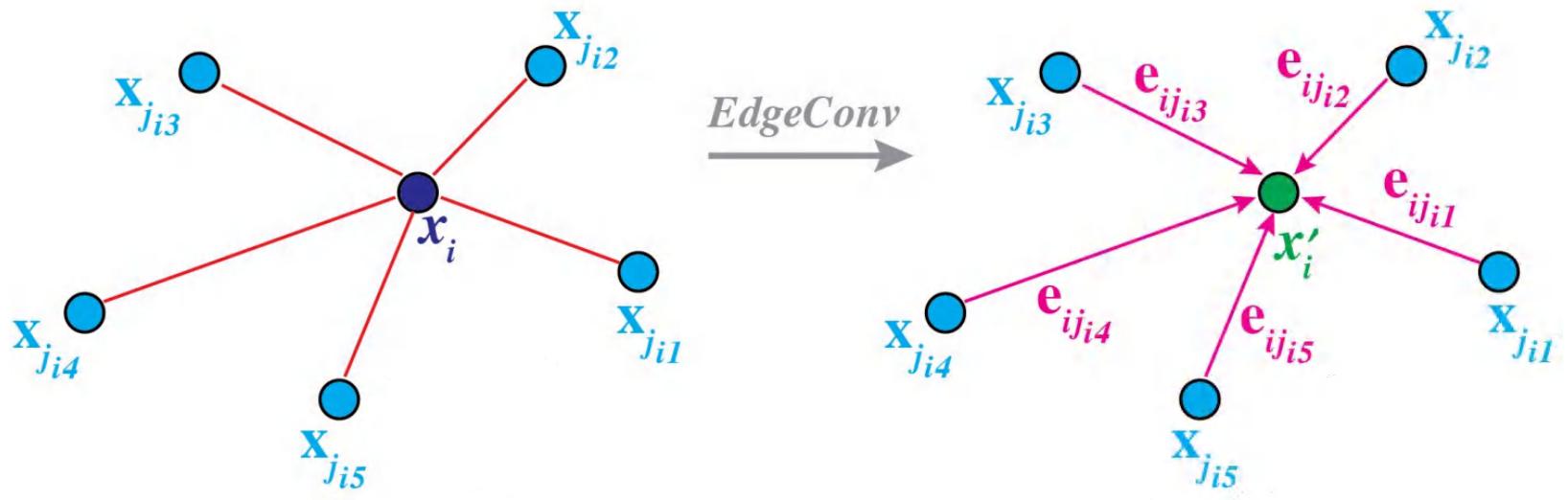


Repeat

- 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

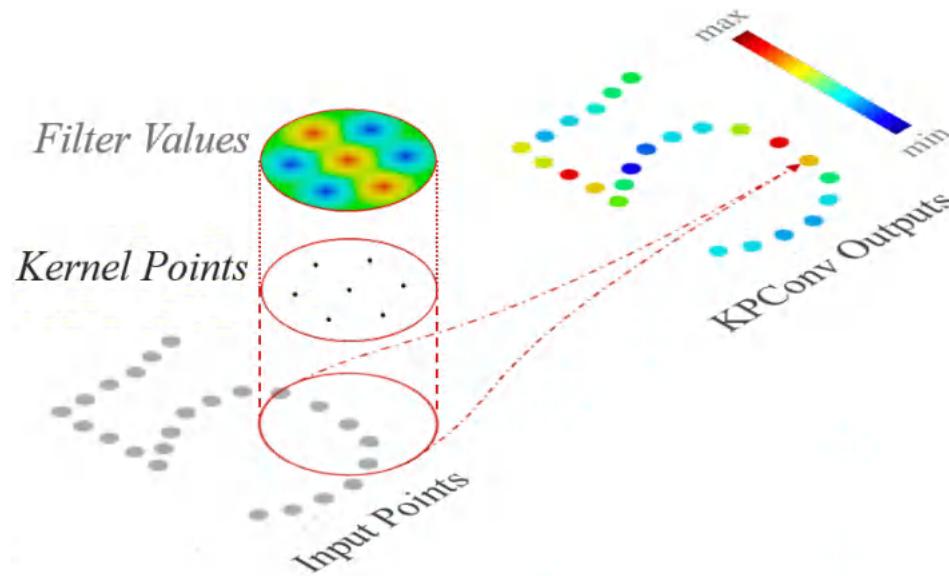
Liu et al., “Relation-Shape Convolutional Neural Network for Point
Cloud Analysis”, CVPR 2019

Issue

- Assume points are sampled from surfaces, the sampling would affect feature extraction :(
- Rescue: Estimate the continuous kernel and point density for continuous convolution

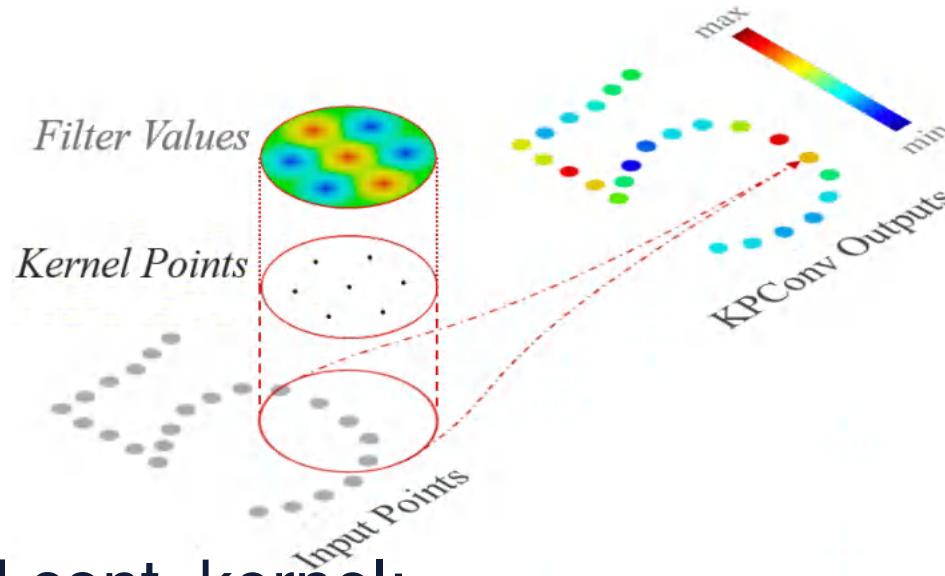
Interpolated Kernel for Convolution

- Continuous conv: $(\mathcal{F} * g)(x) = \int g(y - x)f(y)dy$
- Empirical conv: $(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x)f_i$



Interpolated Kernel for Convolution

- Continuous conv: $(\mathcal{F} * g)(x) = \int g(y - x)f(y)dy$
- Empirical conv: $(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x)f_i$



- Interpolated cont. kernel:

$$\kappa_{jm}(z) = \sum_l k_{ljm} \Phi(|z - y_l|)$$

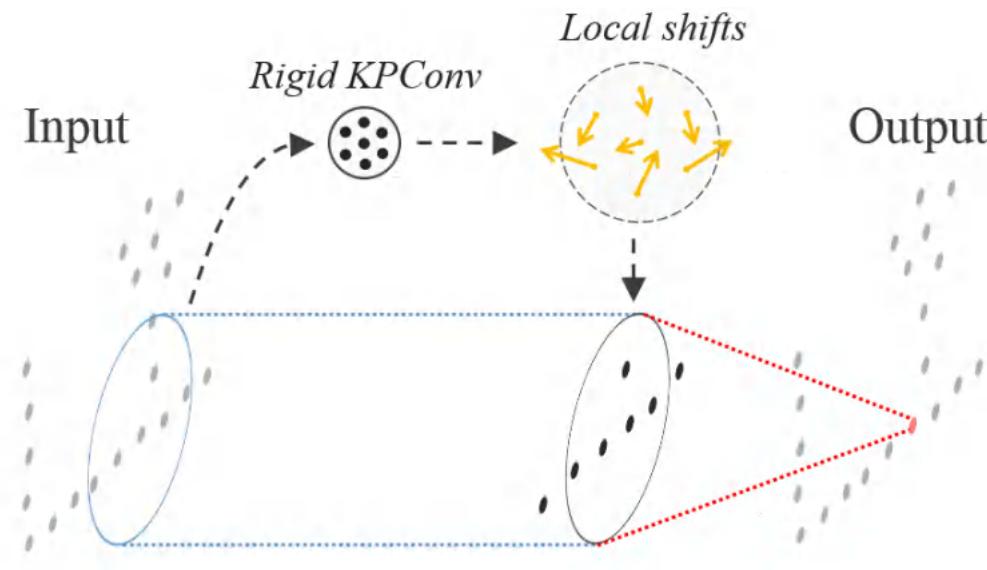
Φ : RBF kernel

Atzmon et al., “**Point Convolutional Neural Networks by Extension Operators**”, *Trans. on Graphics*, 2018

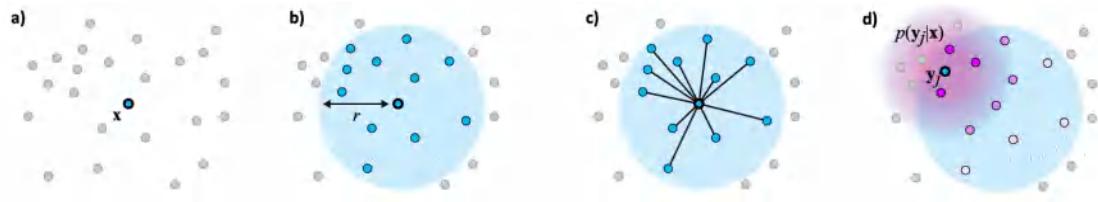
Thomas et al., “**KPConv: Flexible and Deformable Convolution for Point Clouds**”, *ICCV 2019*

Interpolated Kernel for Convolution

- Deformable point-based kernel



Continuous Point Density Estimation



Monte Carlo Integration:

$$\left(\frac{\delta f * g}{\delta \omega_l} \right)(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{j \in \mathcal{N}(\mathbf{x})} \frac{f(\mathbf{y}_j)}{p(\mathbf{y}_j | \mathbf{x})} \frac{\delta g \left(\frac{\mathbf{x} - \mathbf{y}_j}{r} \right)}{\delta \omega_l}$$

RBF Density Estimation:

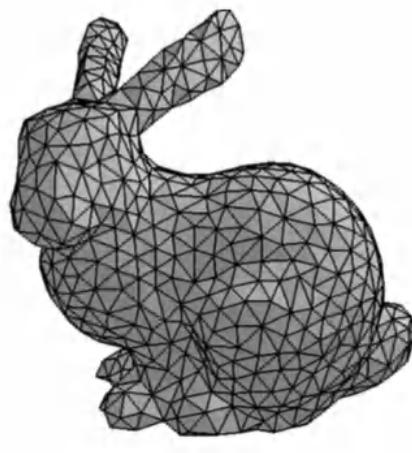
$$p(\mathbf{y}_j | \mathbf{x}) \approx \frac{1}{|\mathcal{N}(\mathbf{x})| \sigma^3} \sum_{k \in \mathcal{N}(\mathbf{x})} \left\{ \prod_{d=1}^3 h \left(\frac{\mathbf{y}_{j,d} - \mathbf{y}_{k,d}}{\sigma} \right) \right\}$$

Spectral Convolution

Shape Processing as Surface Conv

Shapes as surfaces

- Triangle/quad mesh: Piece-wise linear

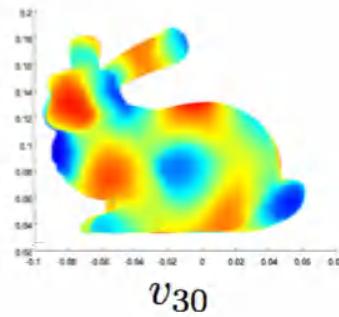
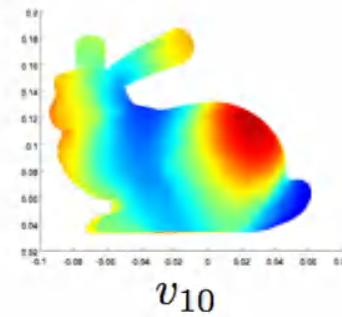
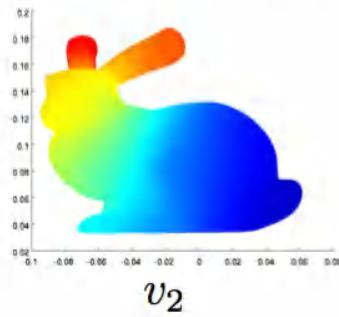


- Coordinates and features as functions defined on the surface (e.g., store at nodes and interpolate in-between)

Fourier Analysis on Surfaces

- Convolution -> linear transformation in the functional space defined on surface
- Bases of the functional space:
 - Eigenfunctions from self-adjoint operators, e.g. Laplacian-Bertrami or Dirac operator

“Fourier basis” of the graph: V : Eigenvectors of Δ



Spectral CNN

- Convolution done in the spectral domain
- Kernels are also built in spectral domain
- Activation done in the spatial domain

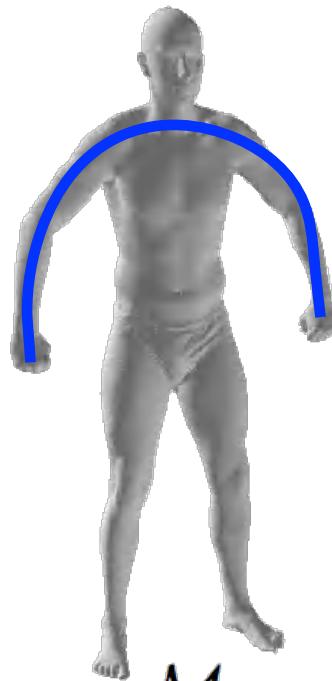
Advantage of Spectral CNN

- Can compare shapes invariant to its embedding
(agnostic to rotation, translation, pose change)

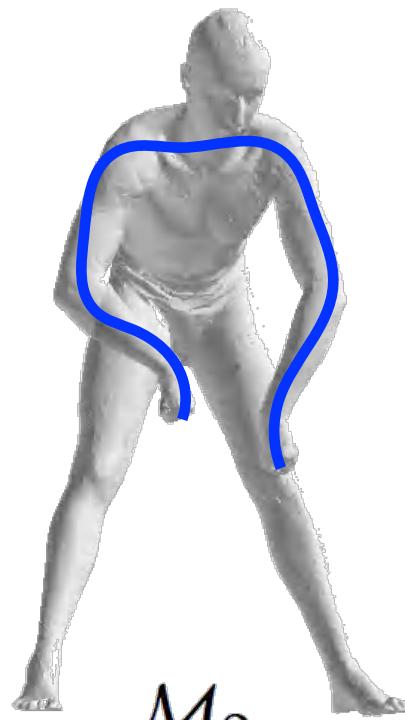


Advantage of Spectral CNN

- Can compare shapes invariant to its embedding
(agnostic to rotation, translation, pose change)



geodesic = intrinsic \mathcal{M}_1



isometry = length-preserving transform \mathcal{M}_2



Advantage of Spectral CNN

- The functional space of a surface under isometric transformation does not change



Visualization of the 5th basis
(Laplacian-Bertrami eigenfunction) at two poses

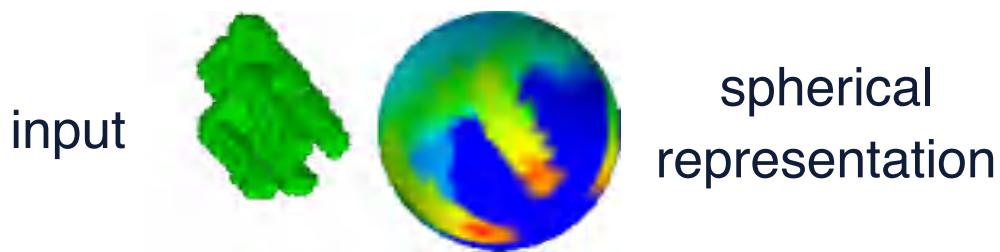
- As a consequence that the Laplacian-Bertrami operator is intrinsic

Fundamental Challenge of Spectral CNN

- If the shapes are not isometric, their spectral domains are not aligned
- Rescue: synchronize them by functional maps

A Special Case: Spherical CNN

- If the surface is always a SPHERE, no worry about the functional space alignment anymore
- Generate a spherical representation



- Do Spectral CNN
 - Has numerical tricks exploiting the symmetry of sphere

Cohen et al., “Spherical CNN”, *ICLR 2018*

Esteves et al., “Learning $SO(3)$ Equivariant Representations with Spherical CNNs”, *ECCV 2018*

A Special Case: Spherical CNN

- Rotation invariance guaranteed

Table 1: ModelNet40 classification accuracy per instance. Spherical CNNs are robust to arbitrary rotations, even when not seen during training, while also having one order of magnitude fewer parameters and faster training.

Method	z/z	SO3/SO3	z/SO3	params	inp. size
PointNet [7]	89.2	83.6	14.7	3.5M	2048 x 3
PointNet++ [38]	89.3	85.0	28.6	1.7M	1024 x 3
VoxNet [29]	83.0	73.0	-	0.9M	30^3
SubVolSup [8]	88.5	82.7	36.6	17M	30^3
SubVolSup MO [8]	89.5	85.0	45.5	17M	20×30^3
MVCNN 12x [9]	89.5	77.6	70.1	99M	12×224^2
MVCNN 80x [9]	90.2	86.0	- ²	99M	80×224^2
RotationNet 20x [30]	92.4	80.0	20.2	58.9M	20×224^2
Ours	88.9	86.9	78.6	0.5M	2×64^2

- Can be used to improve the rot. invariance of MVCNN, as well

Topics

- Classification
- **Segmentation and Detection**
- Reconstruction
- 3D Dataset
- 3D Few-shot Learning

Task: 3D Segmentation & Detection

Object Detection

Input



Output



Object Segmentation

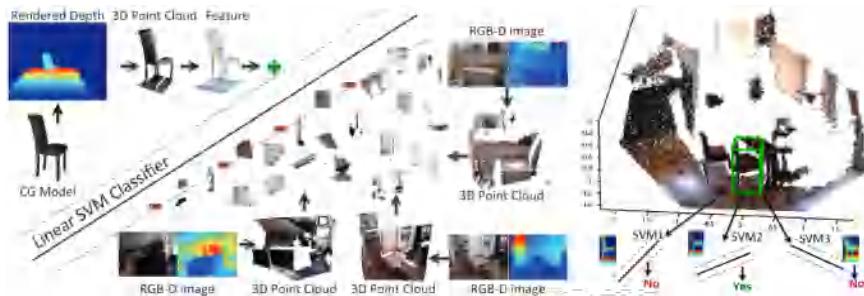


Part Segmentation

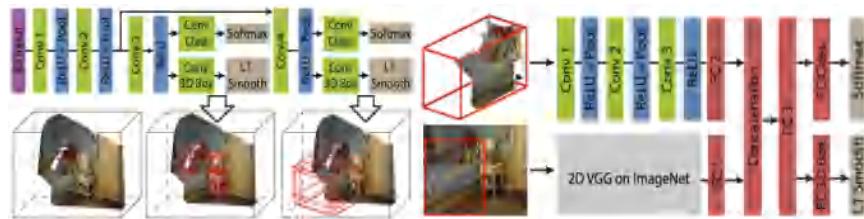


Covered methods: Sliding Shapes, Deep Sliding Shapes, PointRCNN, VoteNet, GSPN, SGPN, Learning to Group

Sliding Shapes



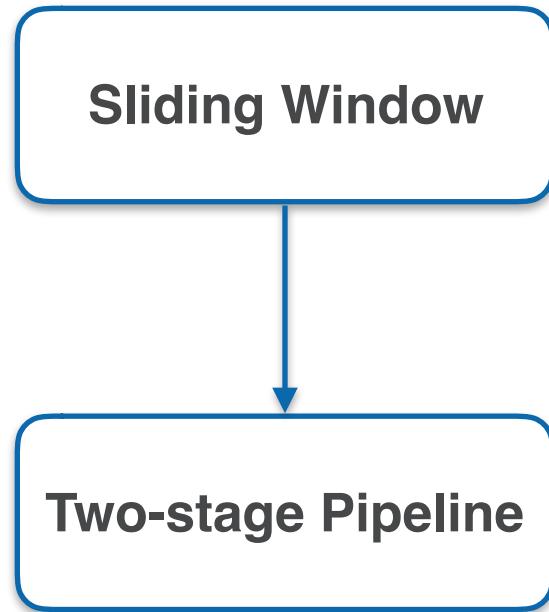
Sliding window to walk over
the entire space



Expensive !

Song et al., “**Sliding Shapes for 3D Object Detection in Depth Images**”, *ECCV 2014*

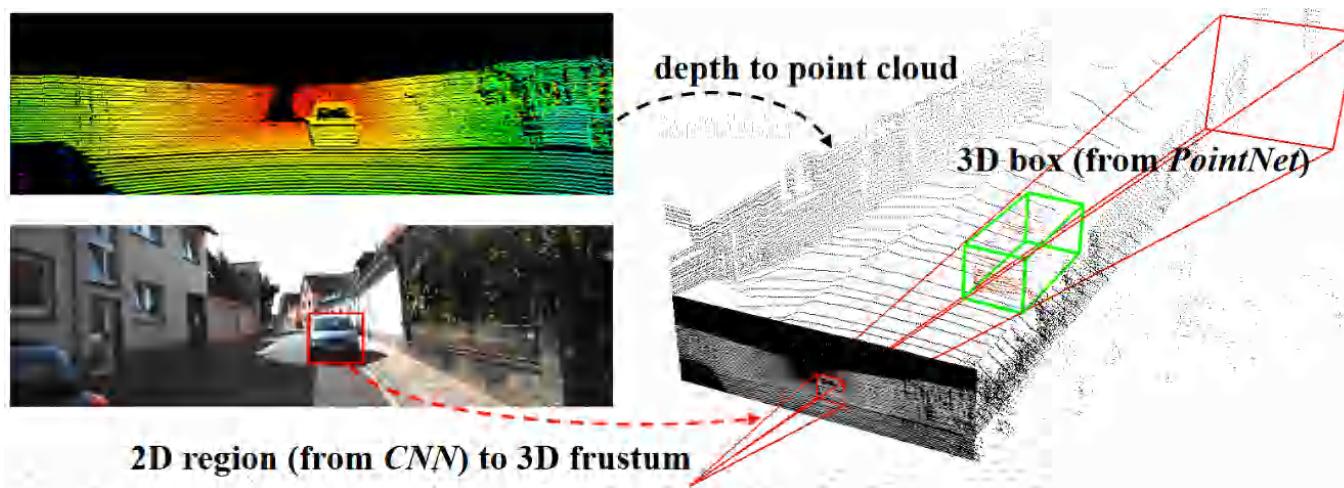
Song et al., “Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images”, *CVPR 2016*



First stage: Proposal
Second stage: Refinement

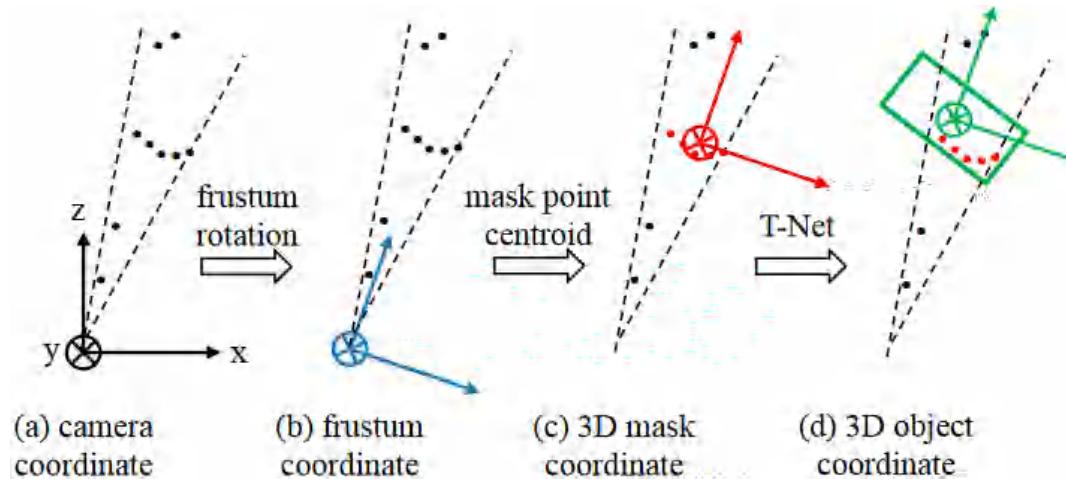
Early Attempt: View-based Proposal

Generate object proposals from a view (e.g., using SSD)



Second-stage: Coordinate Normalization

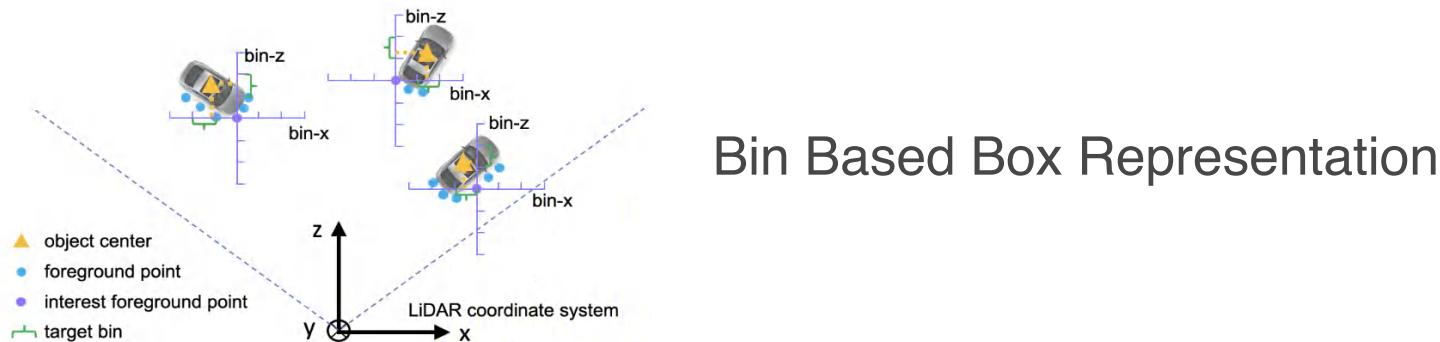
Handle perspective variation in frustum point cloud
by a series of coordinates normalization



Proposal from 3D FG/BG Segmentation

Stage-1: Foreground/Background segmentation to generate 3D proposals

Stage-2: Refine proposals in the canonical coordinates



Proposal from Voting

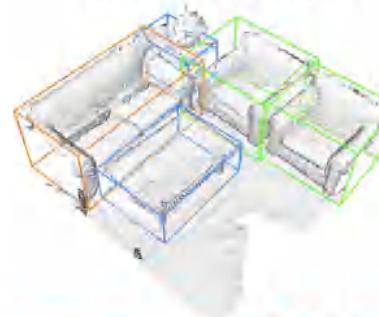
Challenge: 3D object centroid can be far from any surface point, thus hard to regress accurately

- Sample a set of seed points and generate votes, targeting at object centers
- Vote clusters emerge near object centers

Voting from input point cloud

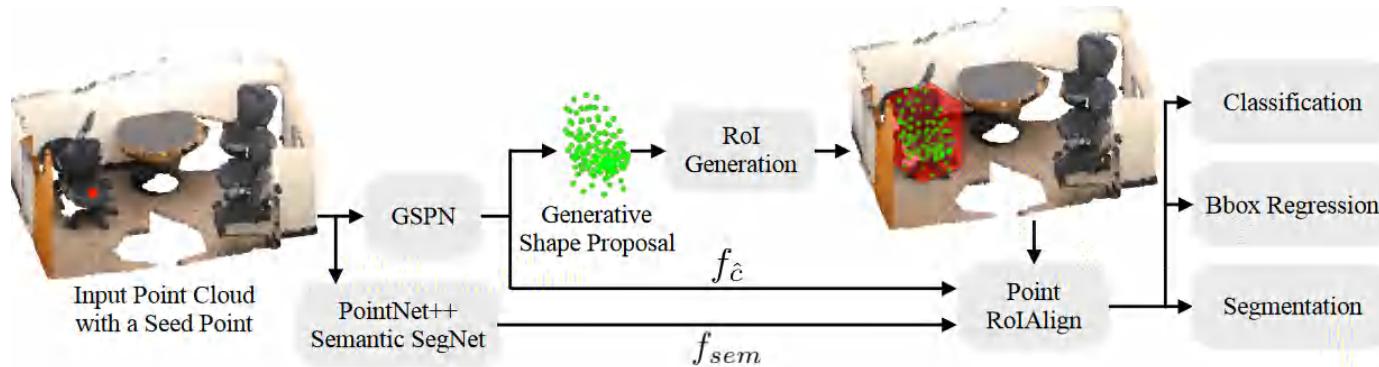


3D detection output



Proposal from Generative Network

- Randomly sample seeds points
- Take point cloud and a seed point as input, use conditional VAE to generate a point cloud as proposal
- Convert the proposal to an ROI box
- R-PointNet (mask RCNN) to segment the object

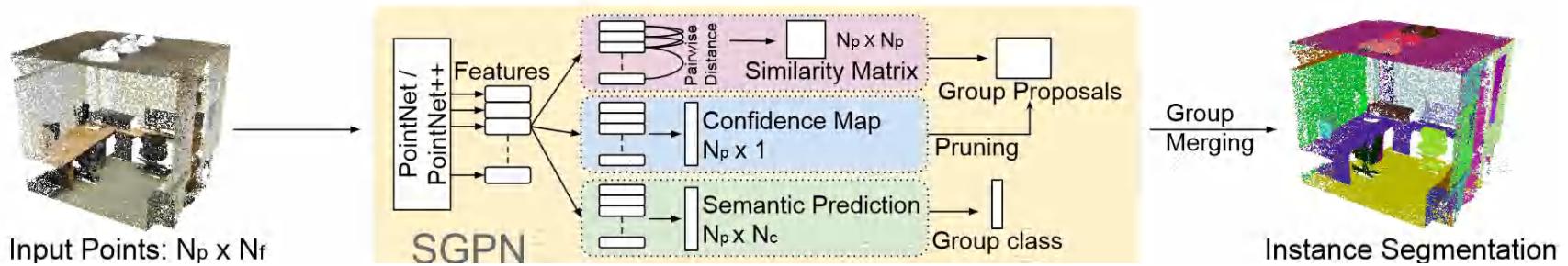


Proposal from Bottom-up Clustering

- Learn a per-point embedding, so that points from the same instance have similar embeddings

$$l(i, j) = \begin{cases} \|F_{SIM_i} - F_{SIM_j}\|_2 & C_{ij} = 1 \\ \alpha \max(0, K_1 - \|F_{SIM_i} - F_{SIM_j}\|_2) & C_{ij} = 2 \\ \max(0, K_2 - \|F_{SIM_i} - F_{SIM_j}\|_2) & C_{ij} = 3 \end{cases}$$

- Clustering gives proposals



- The 3D version of “Associative Embedding”

Wang et al., “SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation”, CVPR 2018

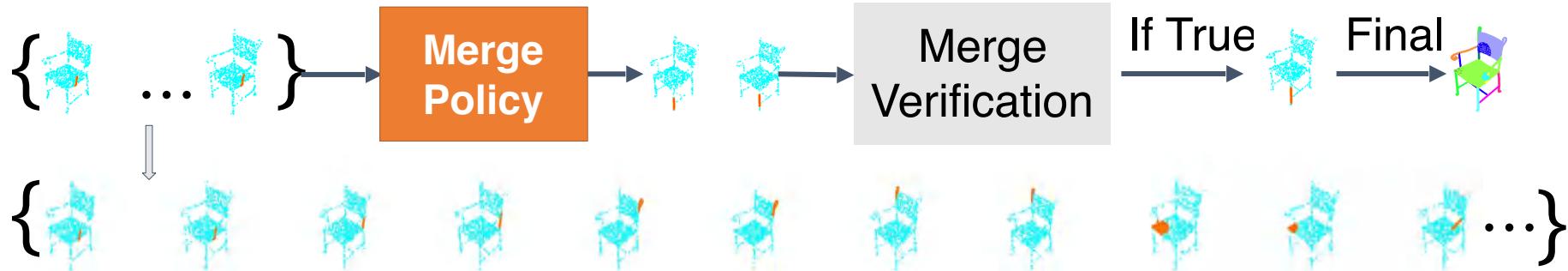
Few-shot Detection

(will be elaborated later)

Learning to Group

- Avoid including context information for generalizability
- Bottom up agglomerative clustering

Sub-Part Pool



Topics

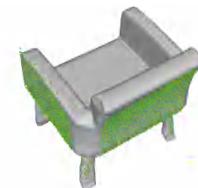
- Classification
- Segmentation and Detection
- Reconstruction
 - Generation Model
 - Multi-View Stereo
- 3D Dataset
- 3D Few-shot/Zero-shot Learning

Task

Conditional generation

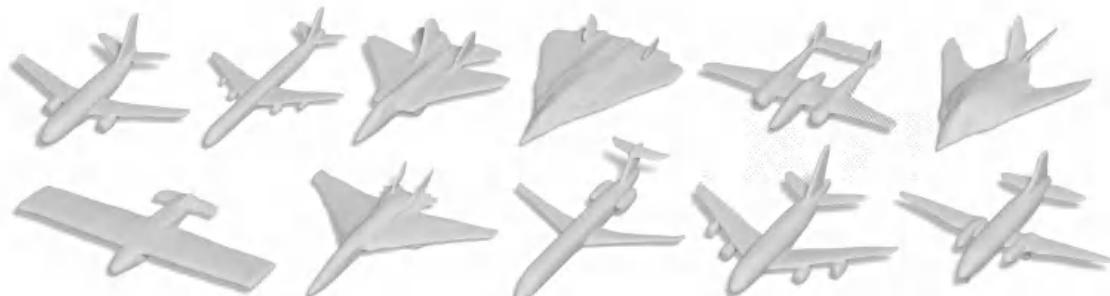


Single-image
3D reconstruction



Shape Completion

Free generation



Gaussian Noise

Metric

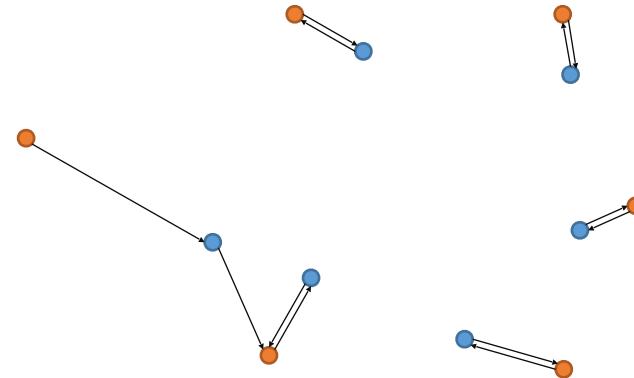
First of all,

how to evaluate the generated shapes?

Metric For Point Clouds

Chamfer Distance

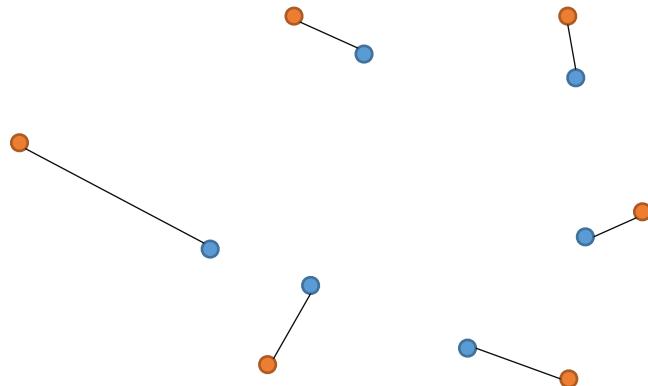
$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$



Earth Mover's Distance

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

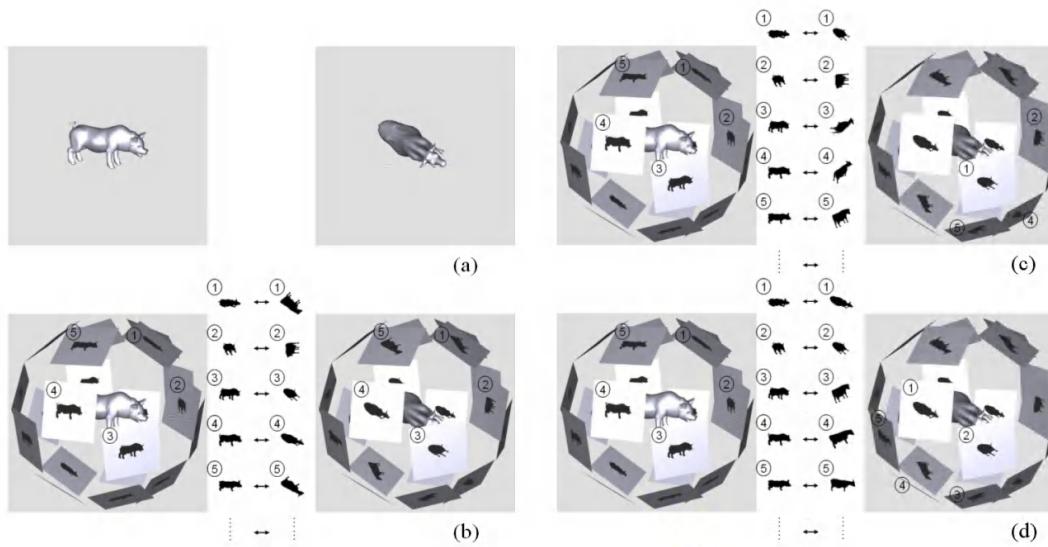
where $\phi : S_1 \rightarrow S_2$ is a bijection.



Metric For Surfaces

Light Field Descriptor (LFD)

- Extract features from orthogonal projections



Chen et al., “On Visual Similarity Based 3D Model Retrieval”,

Computer graphics forum

Chen et al., “Learning Implicit Fields for Generative Shape

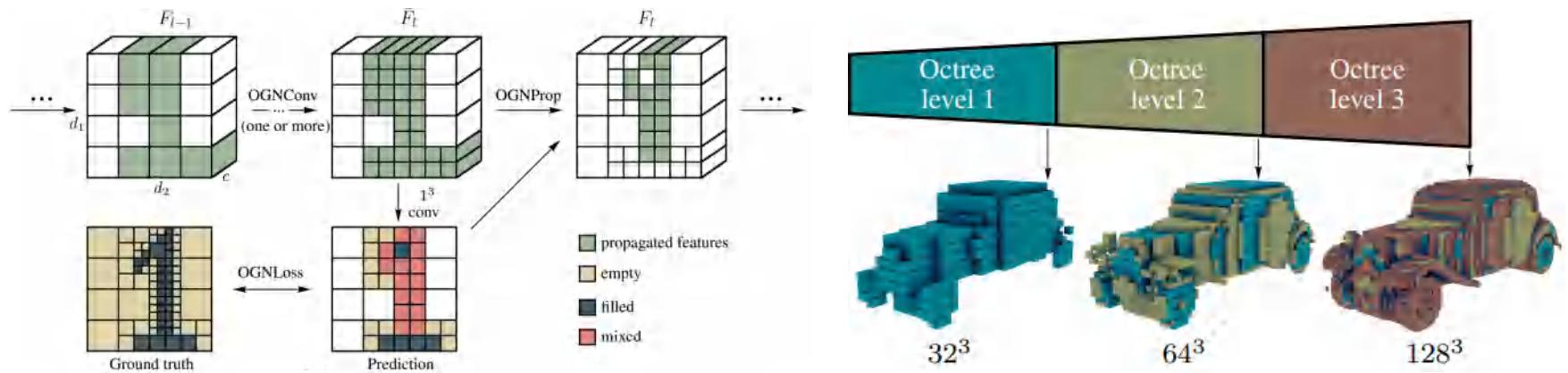
Modeling”, *CVPR 2019*

Algorithm for Conditional Generation

From Single Image to Volume

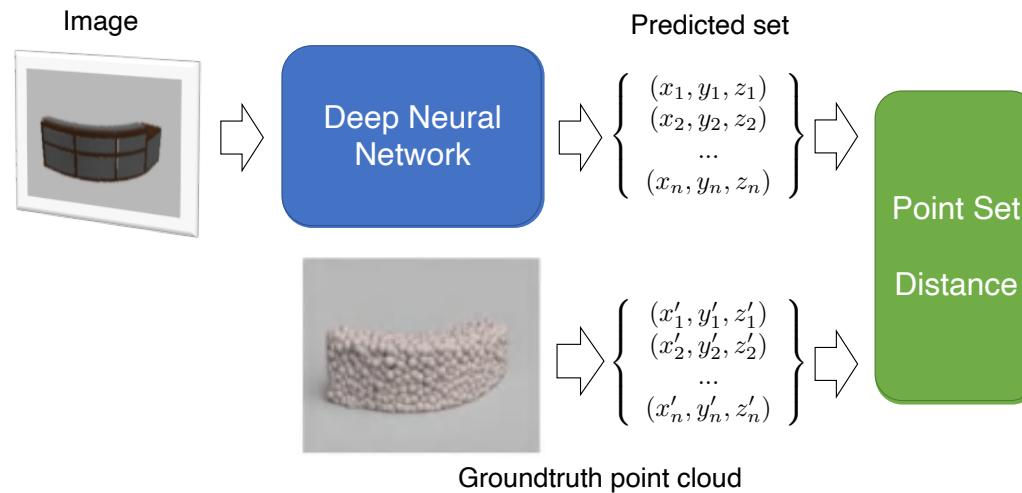
Avoid $\mathcal{O}(n^3)$ reconstruction

- Octree representation of shapes
- Generate the octree layer by layer



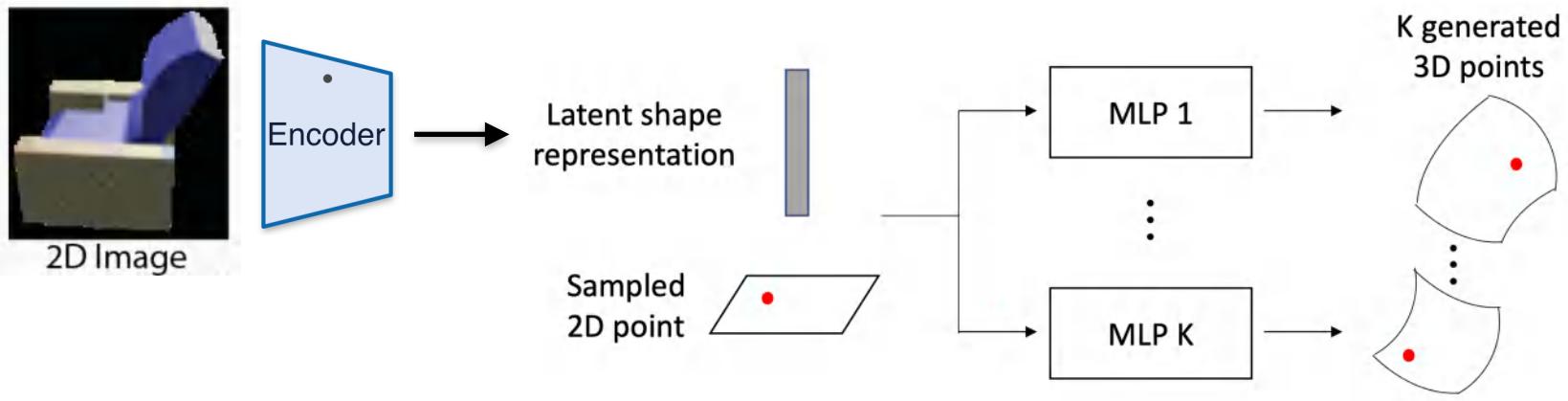
From Single Image to Point Cloud

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



From Image to Surface

- Learn to warp a plane to surface

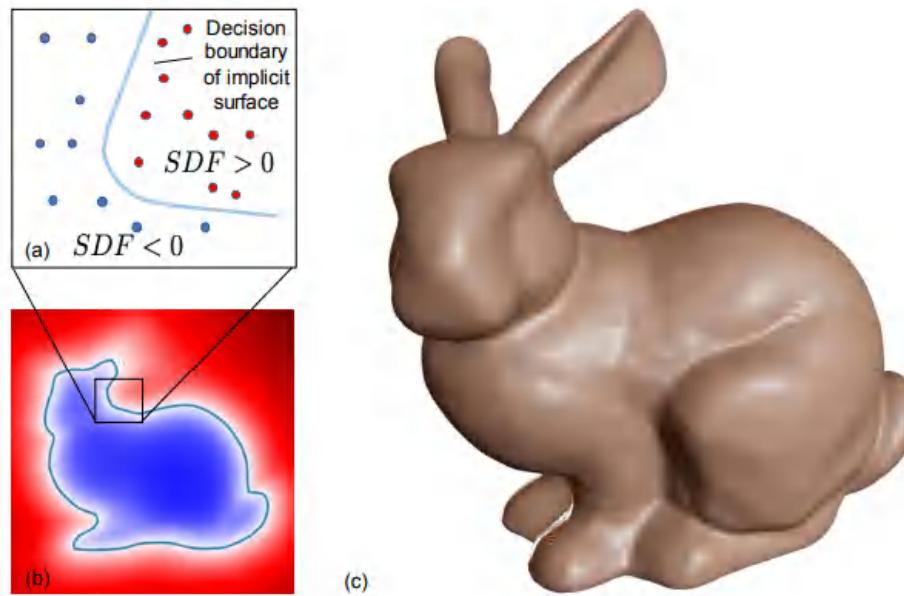


From Image to Surface



Implicit Surface Reconstruction

- Implicit representation of a 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*

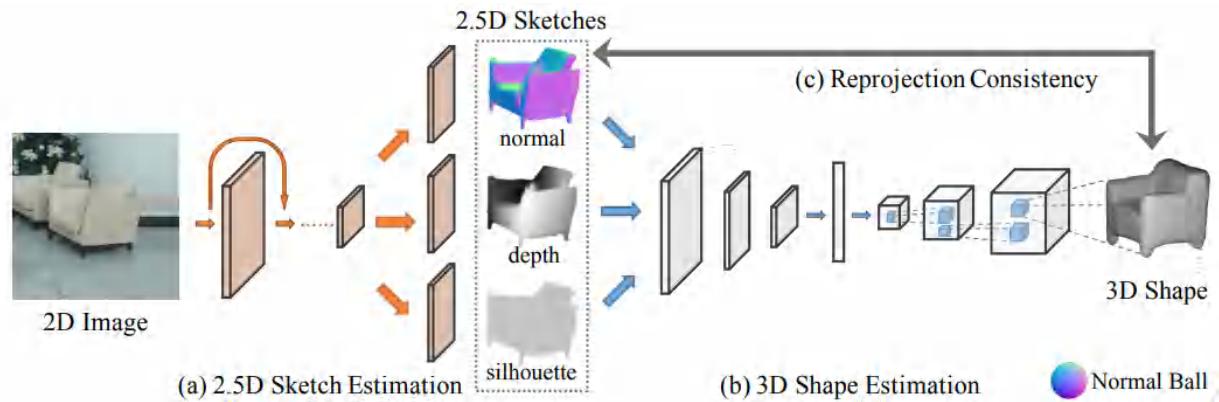
Chen et al., “Learning Implicit Fields for Generative Shape Modeling”, *CVPR 2019*

- In general,
 - First map the input to a shape embedding
 - Then reconstruct by decoding
- Limitation (interpretability)
 - Output is not explicitly grounded on the input
 - Structures of 3D objects not explicitly leveraged

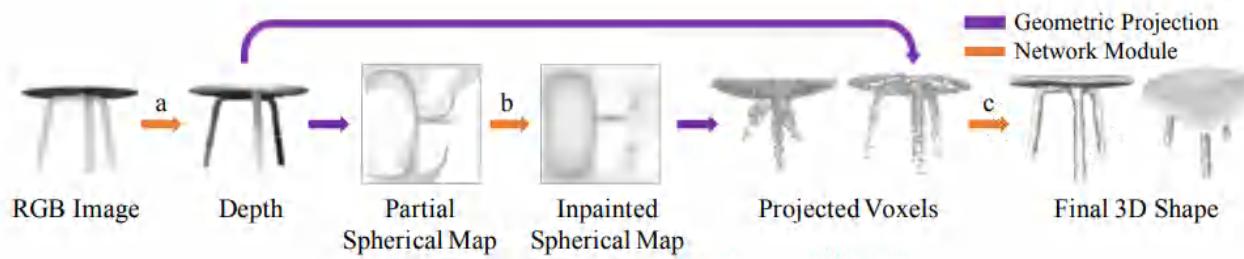
Visually Grounded Prediction: 2.5D to Bridge



Visually Grounded Prediction: 2.5D to Bridge



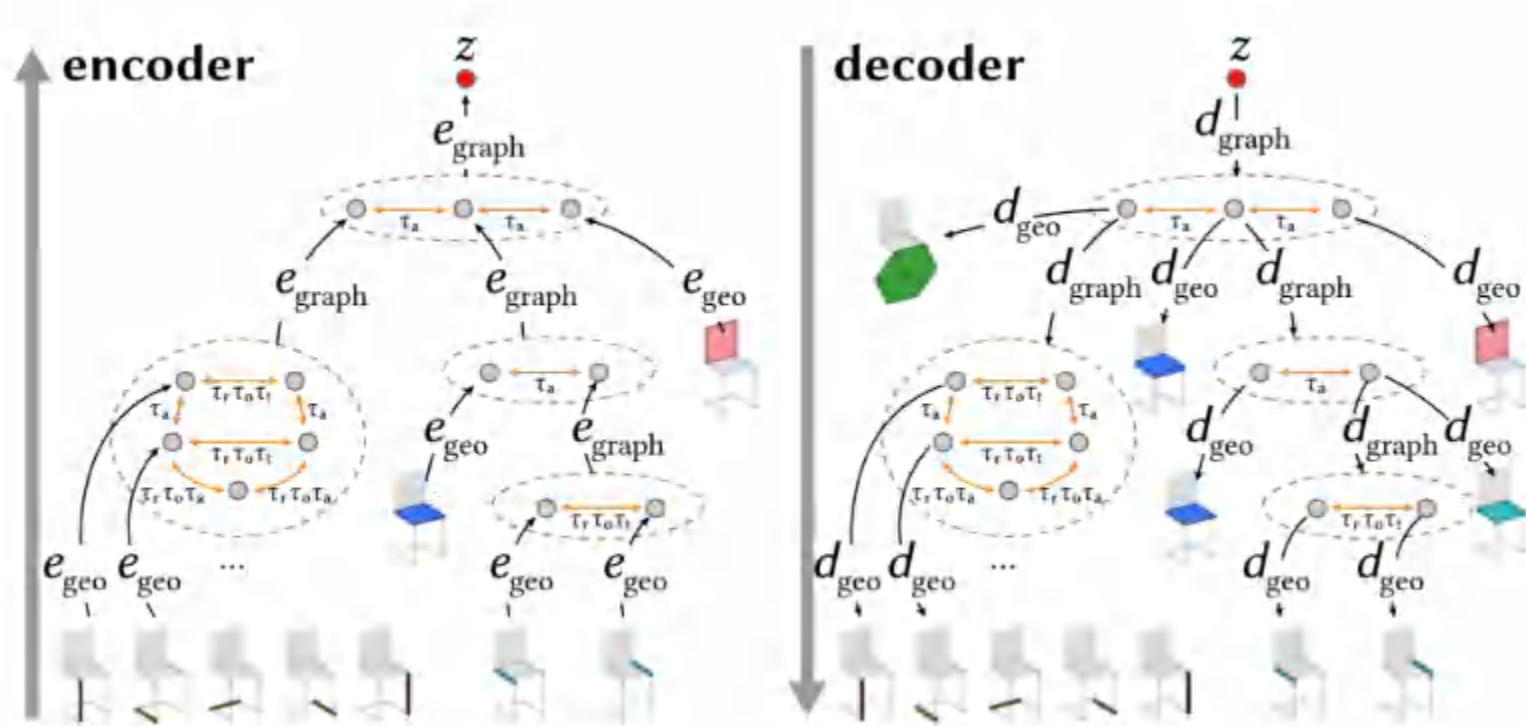
Wu et al., “**MarrNet: 3D Shape Reconstruction via 2.5D Sketches**”, *NeurIPS 2017*



Zhang et al., “**Learning to Reconstruct Shapes from Unseen Classes**”, *NeurIPS 2018*

Structured Prediction: Part-based

Recursive Network for Hierarchical Graph AE



Structured Prediction: Part-based



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

Algorithm for Free Generation (GAN)

Challenges

Similar challenges as GAN for images:

- Good by human eye v.s. Good by objective metric

Metrics

Geometry Quality of Generated Shape

- e.g., MMD for Chamfer/EMD distances

Coverage (COV)

- Model collapse test (The fraction of the shapes in GT dataset that were matched to shapes in generated shapes)

Perceptually Correct

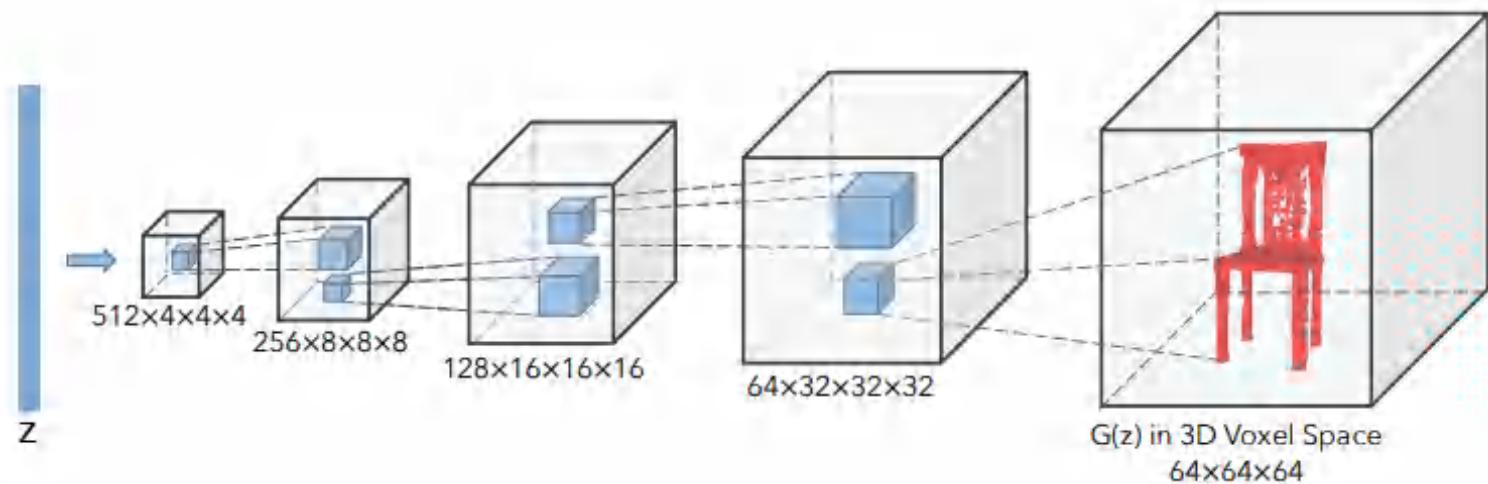
- Feature distribution distance (e.g. Frechet Point Cloud Distance)

$$\text{FPD}(\mathbb{P}, \mathbb{Q}) = \|\mathbf{m}_{\mathbb{P}} - \mathbf{m}_{\mathbb{Q}}\|_2^2 + \text{Tr}(\Sigma_{\mathbb{P}} + \Sigma_{\mathbb{Q}} - 2(\Sigma_{\mathbb{P}}\Sigma_{\mathbb{Q}})^{\frac{1}{2}})$$

Achlioptas et al., “Learning Representations and Generative Models for 3D Point Clouds”, *ICML 2018*

Shu et al., “3D Point Cloud Generative Adversarial Network Based on Tree Structured Graph Convolutions”, *ICCV 2019*

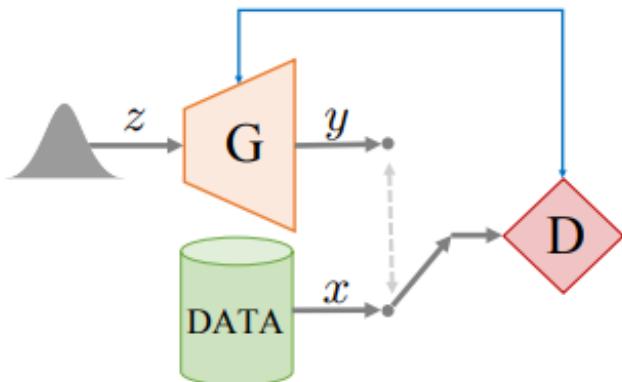
Volumetric Generation



Wu et al., "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling", NeurIPS 2016

Point Cloud Generation

- FC as Generator
- PointNet as Discriminator
- WGAN



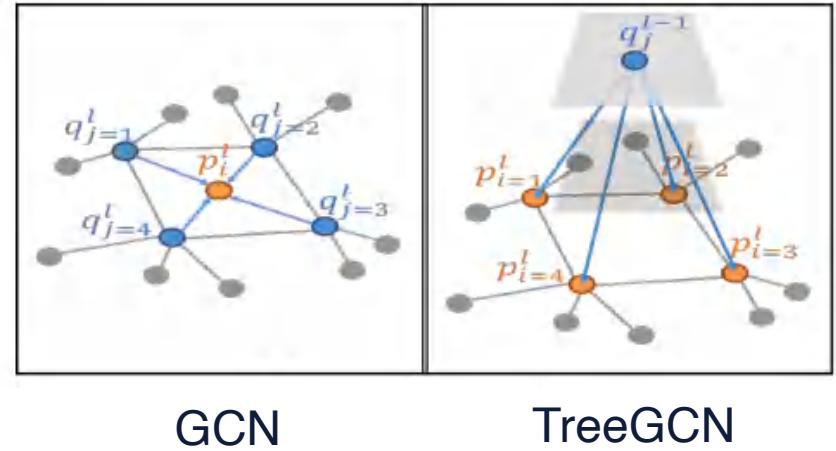
Hierarchical Generation

TreeGAN

- Hierarchical generator
- TreeGCN

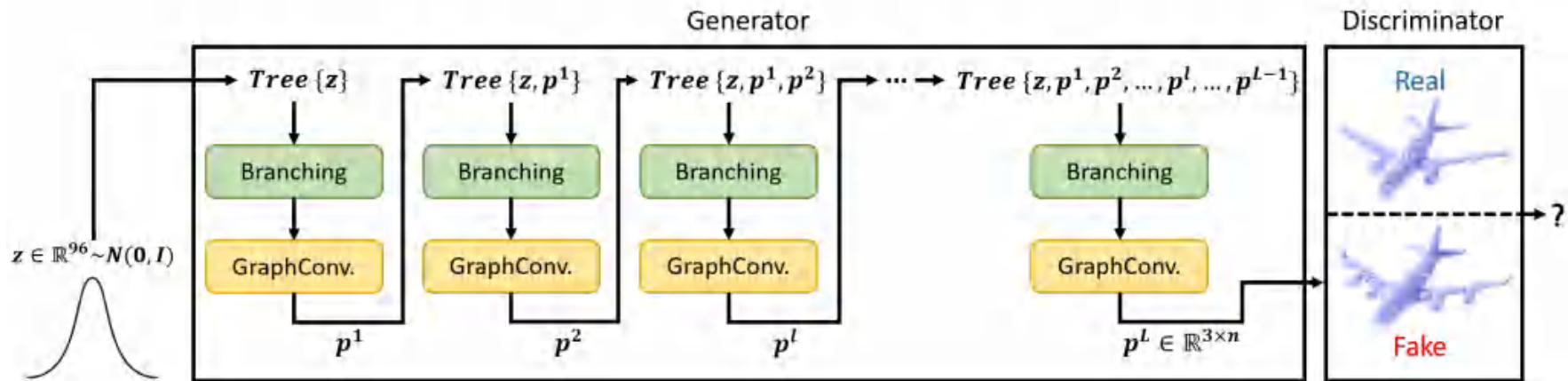
$$p_i^{l+1} = \sigma \left(\mathbf{F}_K^l(p_i^l) + \sum_{q_j \in A(p_i^l)} U_j^l q_j + b^l \right)$$

Ancestor term



GCN

TreeGCN



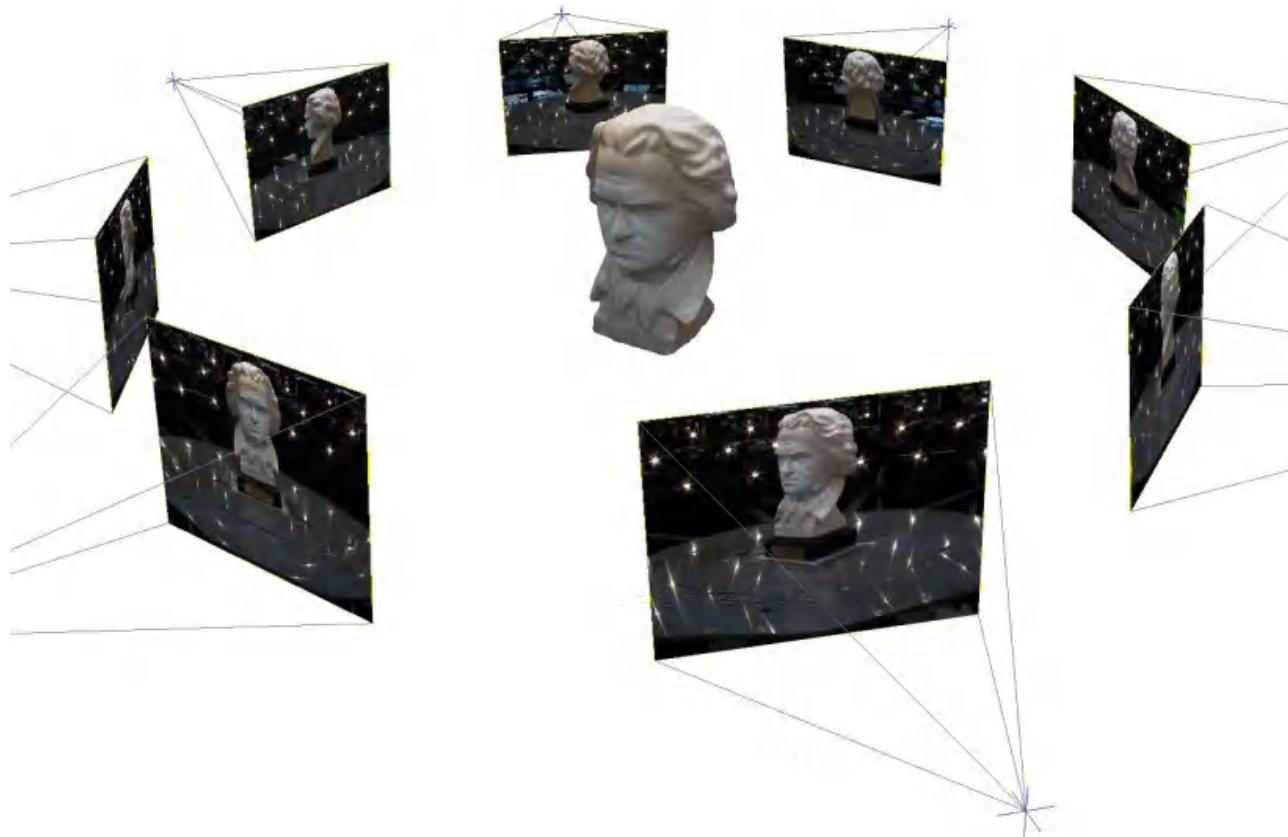
Many Issues

- Still cannot generate high quality local details
- Still hard to generate complex structures
- Use stronger classifiers (than PointNet) for
discriminator is highly tricky

Topics

- Classification
- Segmentation and Detection
- Reconstruction
 - Generation Model
 - Multi-View Stereo
- 3D Dataset
- 3D Few-shot/Zero-shot Learning

Task: Reconstruction

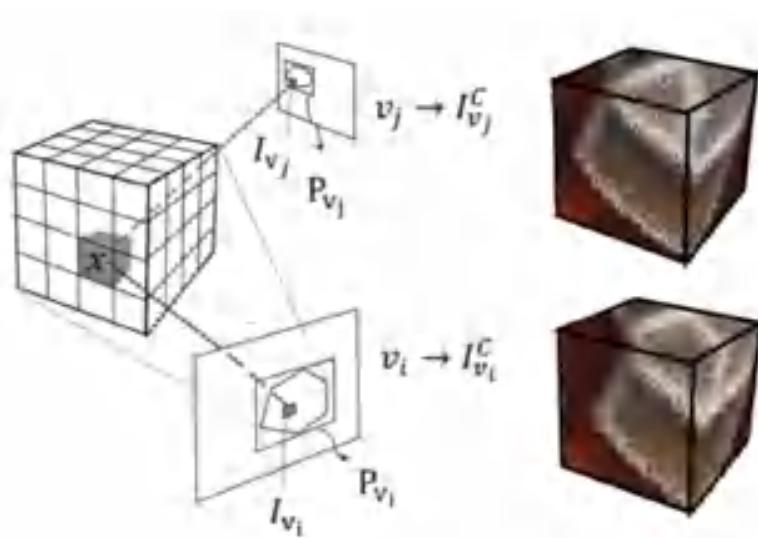


[image: oswald]

Covered methods: SurfaceNet, LSM, GC-Net, MVSNet, R-MVSNet, PointMVSNet, BA-Net

Surface Reconstruction as Voxel Occupancy Prediction

Unprojection along viewing rays to build colored voxel cubes.



Predict the surface confidence for each voxel:

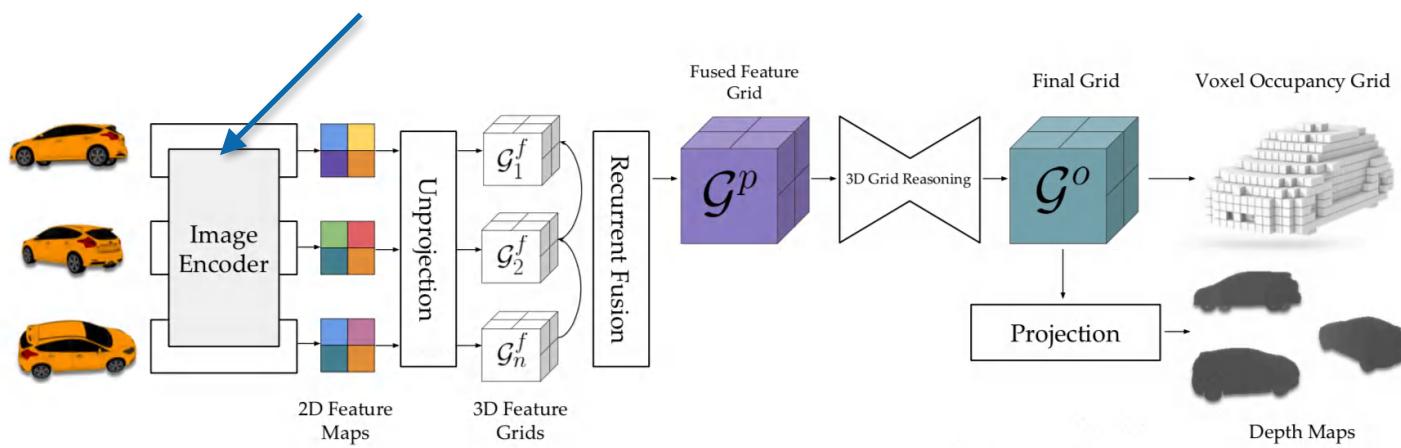
$$\begin{aligned} L(I_{v_i}^C, I_{v_j}^C, \hat{S}^C) = \\ - \sum_{x \in C} \{\alpha \hat{s}_x \log p_x + (1 - \alpha)(1 - \hat{s}_x) \log(1 - p_x)\} \end{aligned}$$

Limitations:

- Pre-computed grids can only take RGB colors at coarse resolution
- Voxel binarization introduces quantization errors.

Learning-Based Stereopsis

- End-to-end learning of deep features for each pixel.

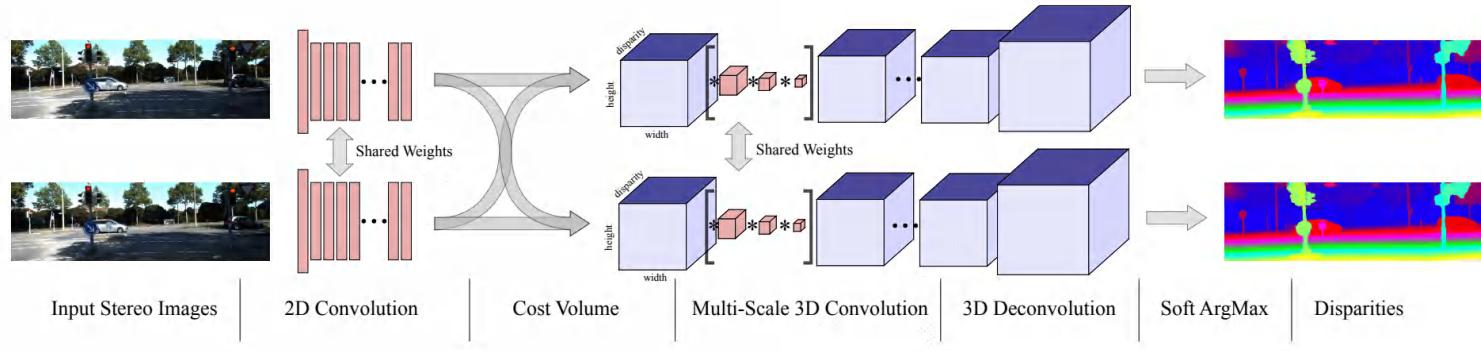




Still very coarse resolution (32x32x32)
due to volumetric representation.

Improve Output Resolution

- Differentiable soft-argmin to achieve sub-pixel accuracy.

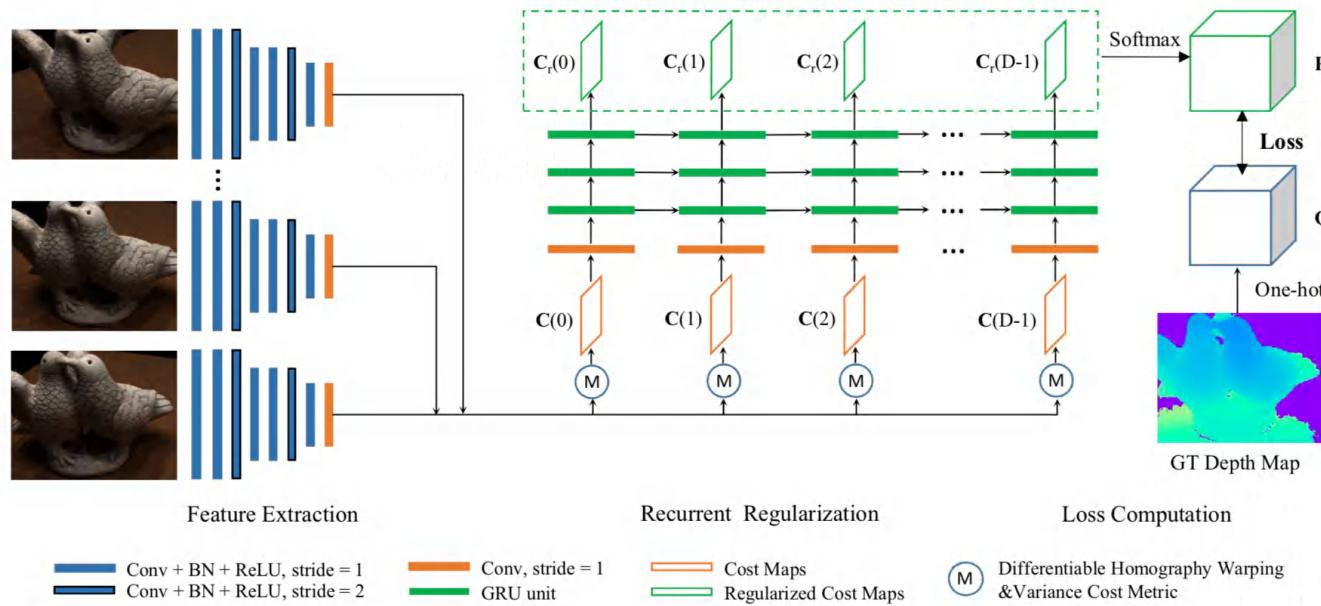


$$\text{soft argmin} := \sum_{d=0}^{D_{\max}} d \times \sigma(-c_d)$$

- View-aligned cost-volume construction.

Input Input Resolution

Idea 1: Slide-by-Slide Processing of Cost Volume by Recurrent Neural Network



The cost volume is sequentially regularized along the depth direction.

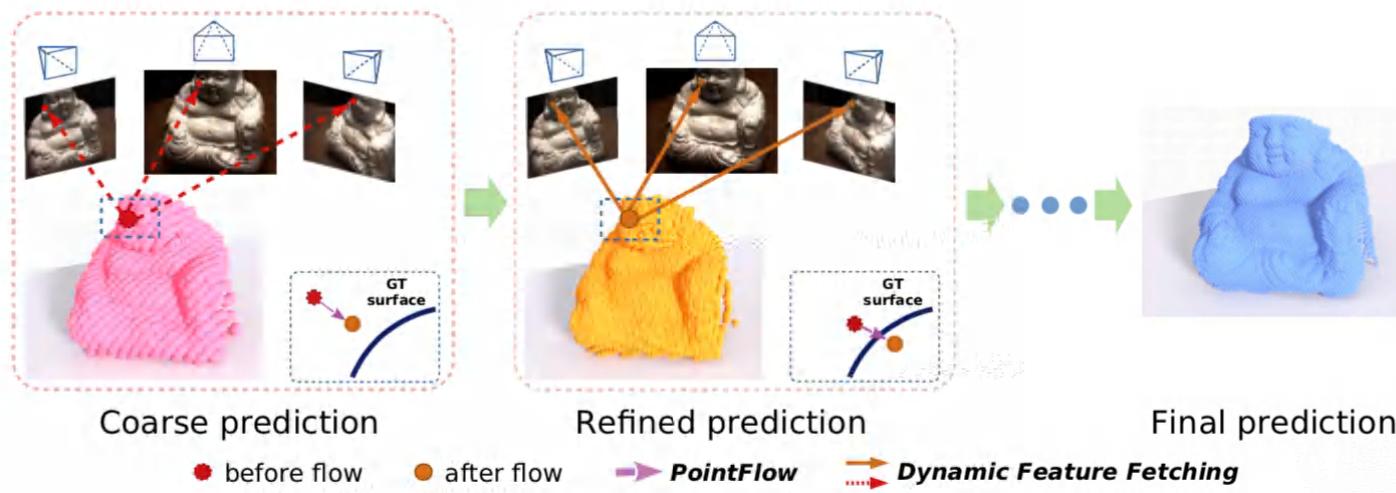
Yao et al., “**MVSNet: Depth Inference for Unstructured Multi-view Stereo**”, *ECCV 2018*

Yao et al., “**Recurrent MVSNet for High-resolution Multi-view Stereo Depth Inference**”, *CVPR 2019*

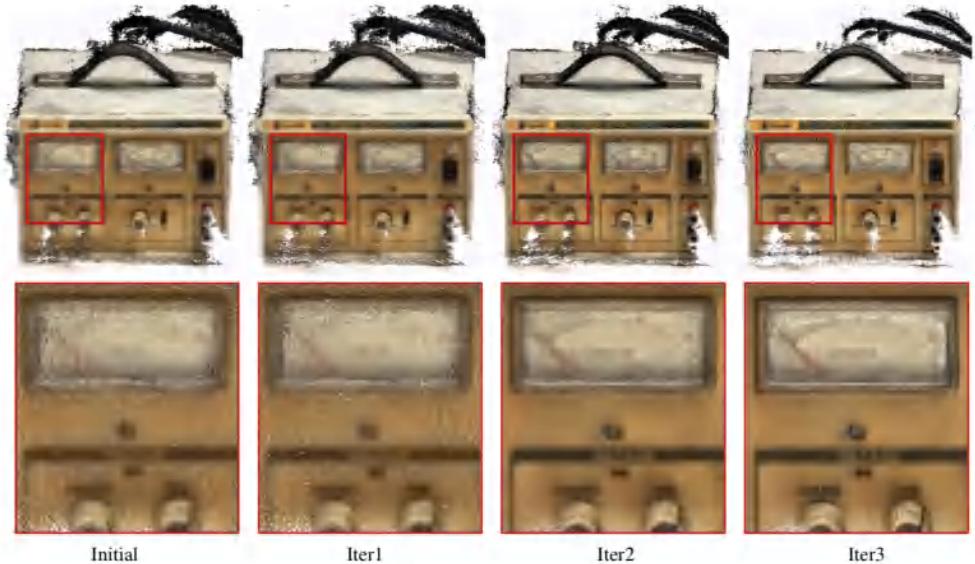
Input Input Resolution

Idea 2: Point-based MVS

- Point-based representation for computational efficiency.
- Iteratively update the location of points and spawn more points.
- More flexible and accurate.



Iterative refinement:



Results on DTU benchmark

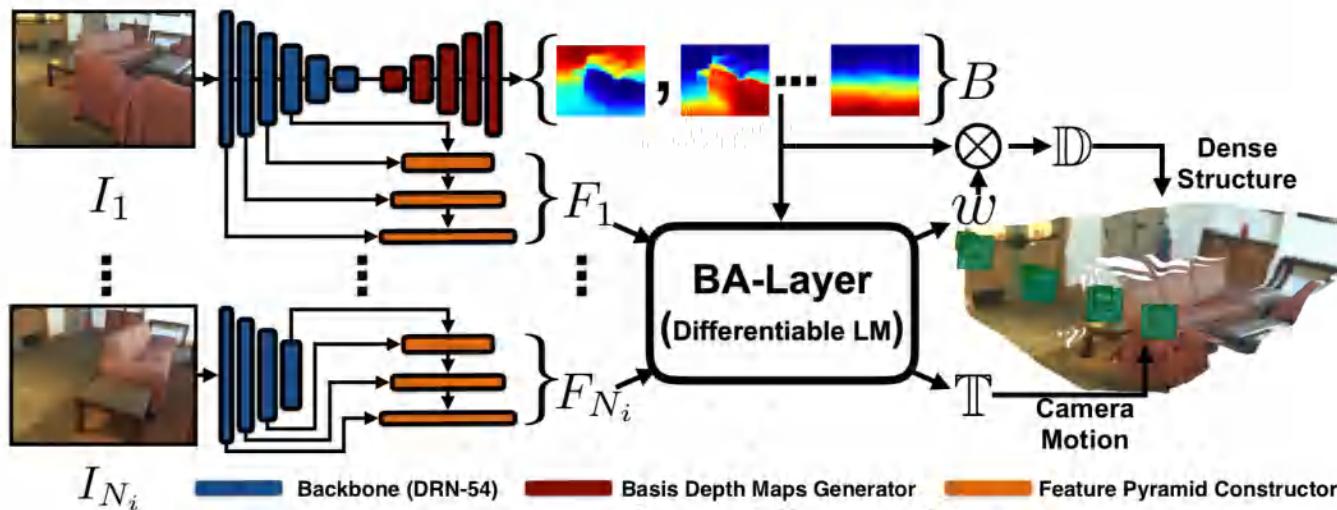
Iter.	Acc. (mm)	Comp. (mm)	Overall (mm)	0.5mm <i>f-score</i>	Depth Map Res.	Depth Interval (mm)	GPU Mem. (MB)	Runtime (s)
-	0.693	0.758	0.726	47.95	160×120	5.30	7219	0.34
1	0.674	0.750	0.712	48.63	160×120	5.30	7221	0.61
2	0.448	0.487	0.468	76.08	320×240	4.00	7235	1.14
3	0.361	0.421	0.391	84.27	640×480	0.80	8731	3.35
MVSNet[29]	0.456	0.646	0.551	71.60	288×216	2.65	10805	1.05

Learning for SfM

- Above learning-based MVS methods all **assume relative camera pose**
- What if not?
 - Classic 3D: Bundle Adjustment
- Learning-based bundle adjustment

BA-Net

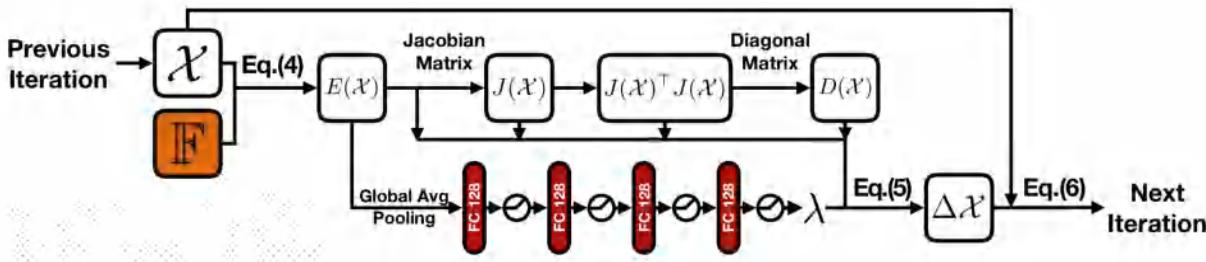
End-to-end pipeline for SfM with differentiable bundle adjustment.



Differentiable LM algorithm:

- Iterative update as rollout of network layers
- Use network to predict the damping factor lambda.

BA-layer:



$$\Delta \mathcal{X} = (J(\mathcal{X})^\top J(\mathcal{X}) + \lambda D(\mathcal{X}))^{-1} J(\mathcal{X})^\top E(\mathcal{X}).$$

Topics

- Classification
- Segmentation and Detection
- Reconstruction
- **3D Dataset**
- 3D Few-shot Learning

Datasets for 3D Scenes

Large-scale Synthetic Objects: ShapeNet



3DScan: Consumer-grade 3D scanning (click to open)

ModelNet: absorbed by ShapeNet

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

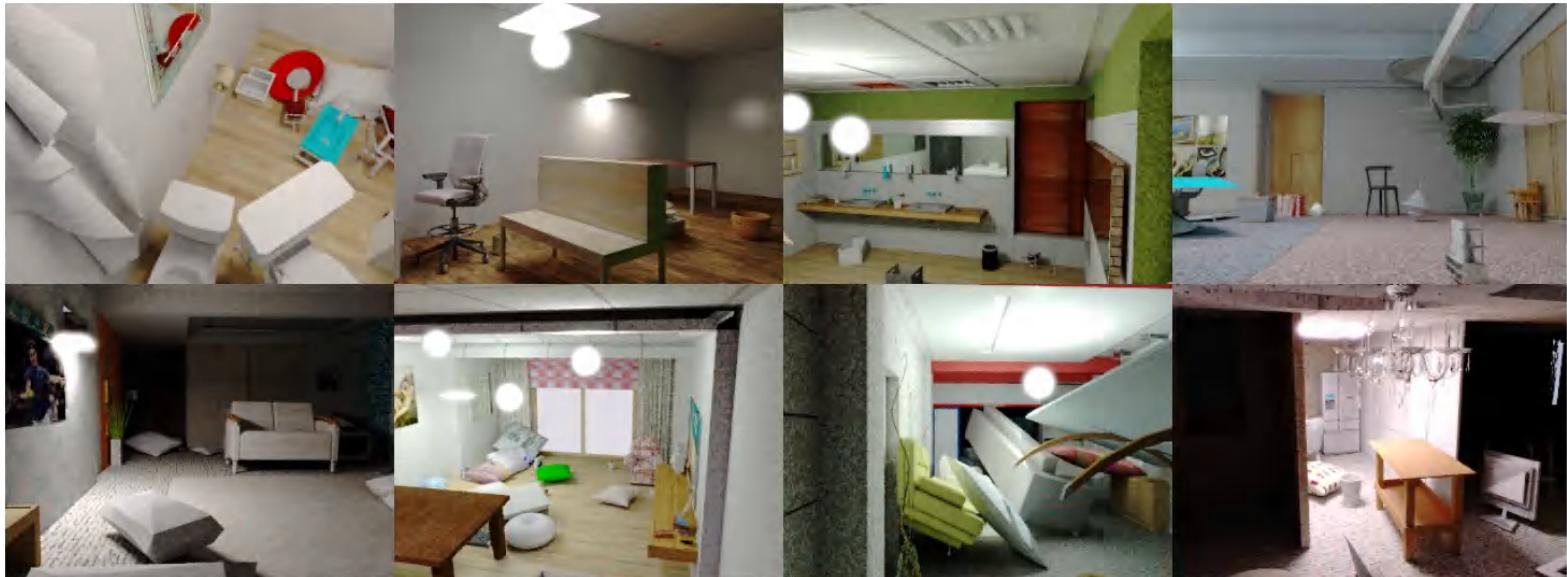
Wu et al., "3D ShapeNets: A deep representation for volumetric shapes", *CVPR 2015*

Choi et al., "A Large Dataset of Object Scans", *arXiv*

Datasets for 3D Scenes

Large-scale Synthetic Scenes: SceneNet

- 3D meshes
- 5M Photorealistic Images



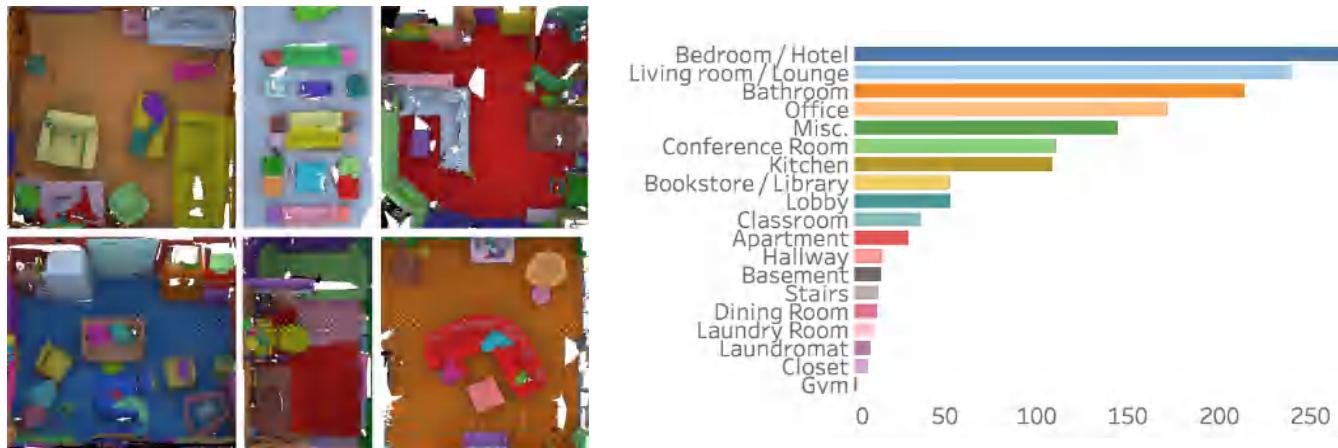
Ankur et al., “Understanding RealWorld Indoor Scenes with Synthetic Data”, *CVPR 2016*

McCormac et al., “SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?”, *ICCV 2017*

Datasets for 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



Datasets for 3D Object Parts

Coarse-grained Part: ShapeNetPart2016

airplane (4027)



wings body
tail engine

earphone (73)



headband
earphone

cap (56)



peak panel

motorbike (336)



wheel
light gas tank
seat handle

guitar (793)



body
head
neck

knife (420)



handle blade

rocket (85)



fin
body
nose

lamp (2308)



shade
base
tube

chair (6742)



seat
back
arm
leg

pistol (307)



handle
barrel
trigger

car (7496)



wheels
roof hood

skateboard (152)

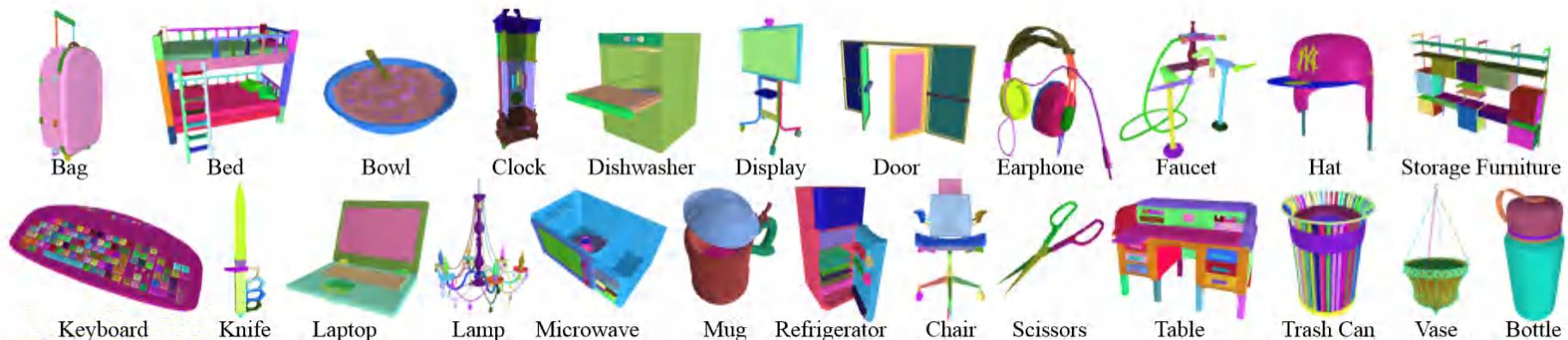


deck wheel

Datasets for 3D Object Parts

Fine-grained Part: PartNet (ShapeNetPart2019)

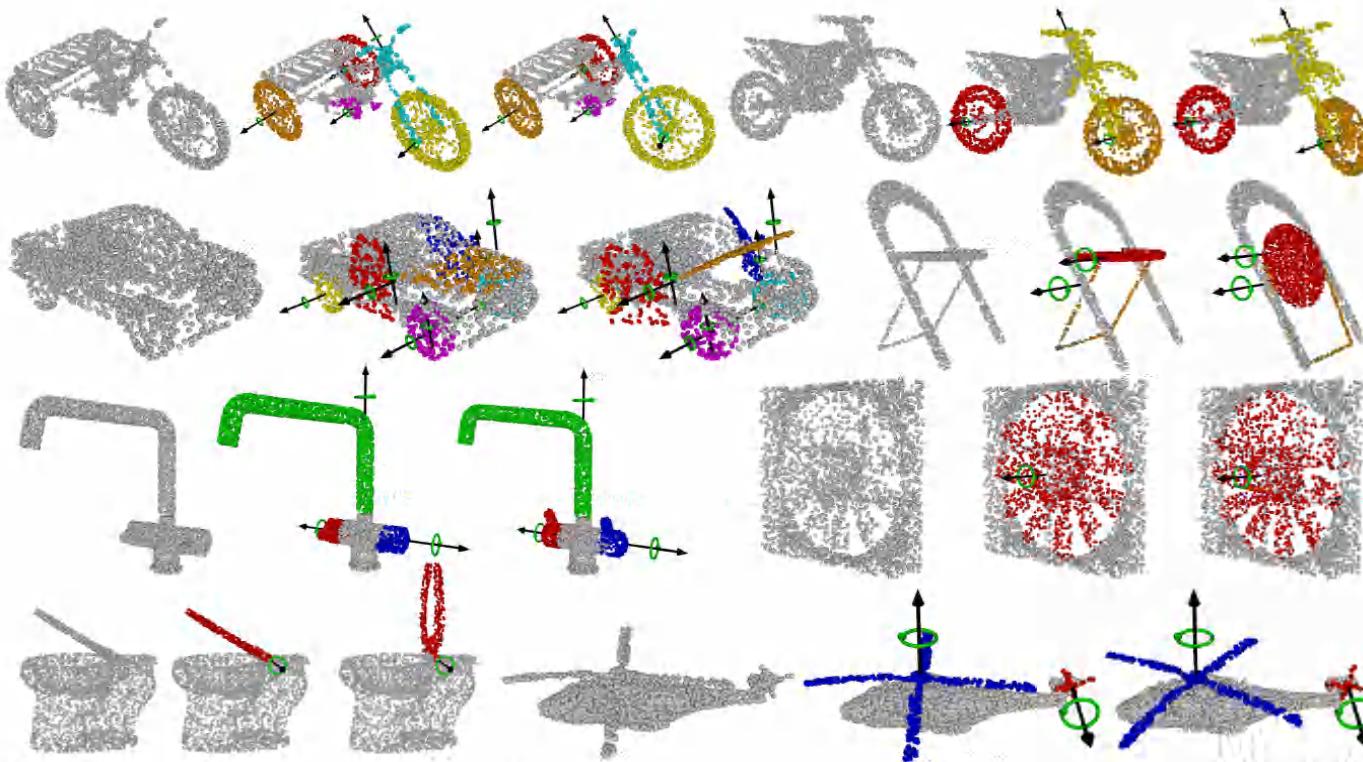
- Fine-grained (towards mobility)
- Instance-level
- Hierarchical



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

Dataset for Object Part Motion

Shape2Motion Dataset



Mobility Analysis of 3D Shapes

Topics

- Classification
- Segmentation and Detection
- Reconstruction
- 3D Dataset
- **3D Few-shot/Zero-shot Learning**

Why Few-shot/Zero-shot Learning by 3D?

- Can be a better platform than images
- Shapes are **pure** and **complete**
 - No contamination by distortion, illumination, viewpoint change, ...

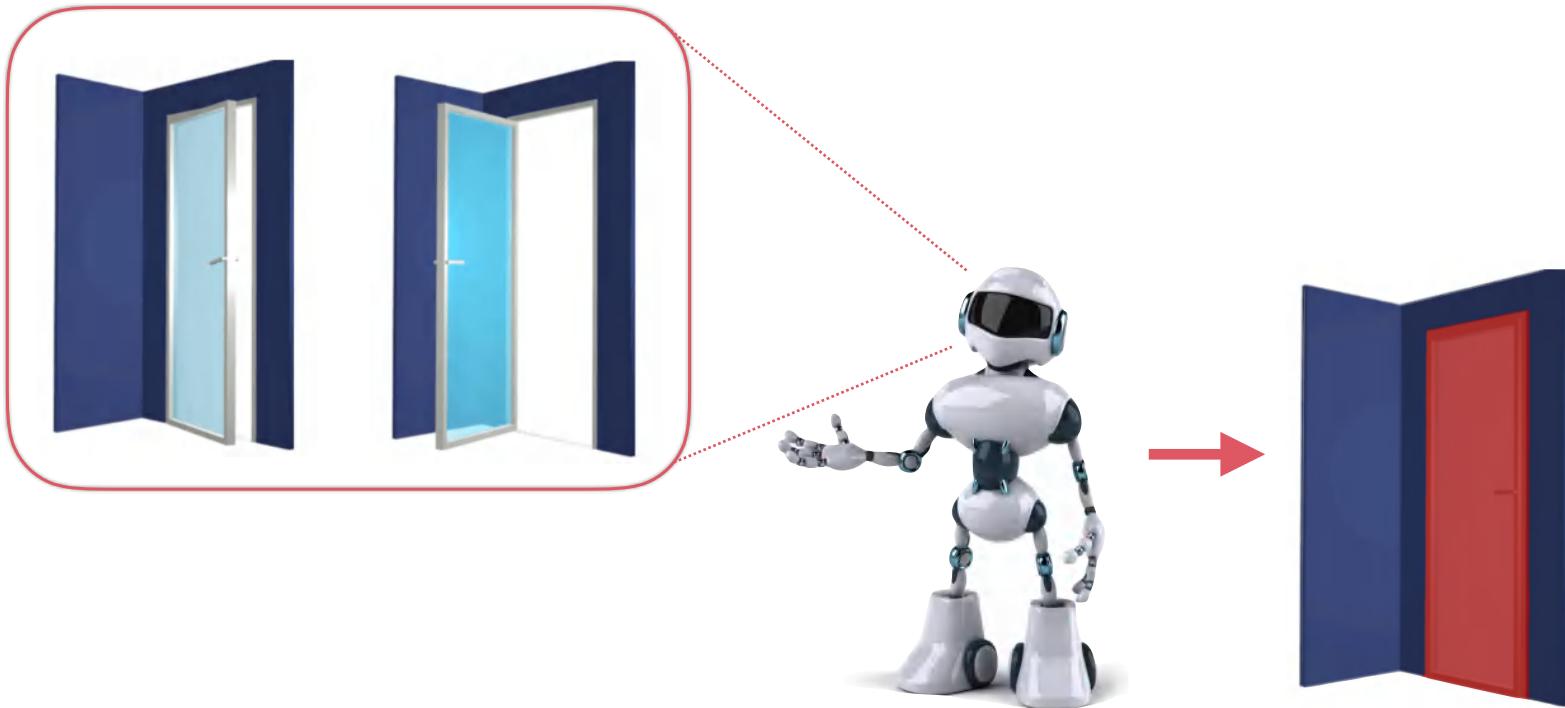
If one image is more than a thousand words, then one shape is more than one thousand pictures

Why Few-shot/Zero-shot Learning by 3D?

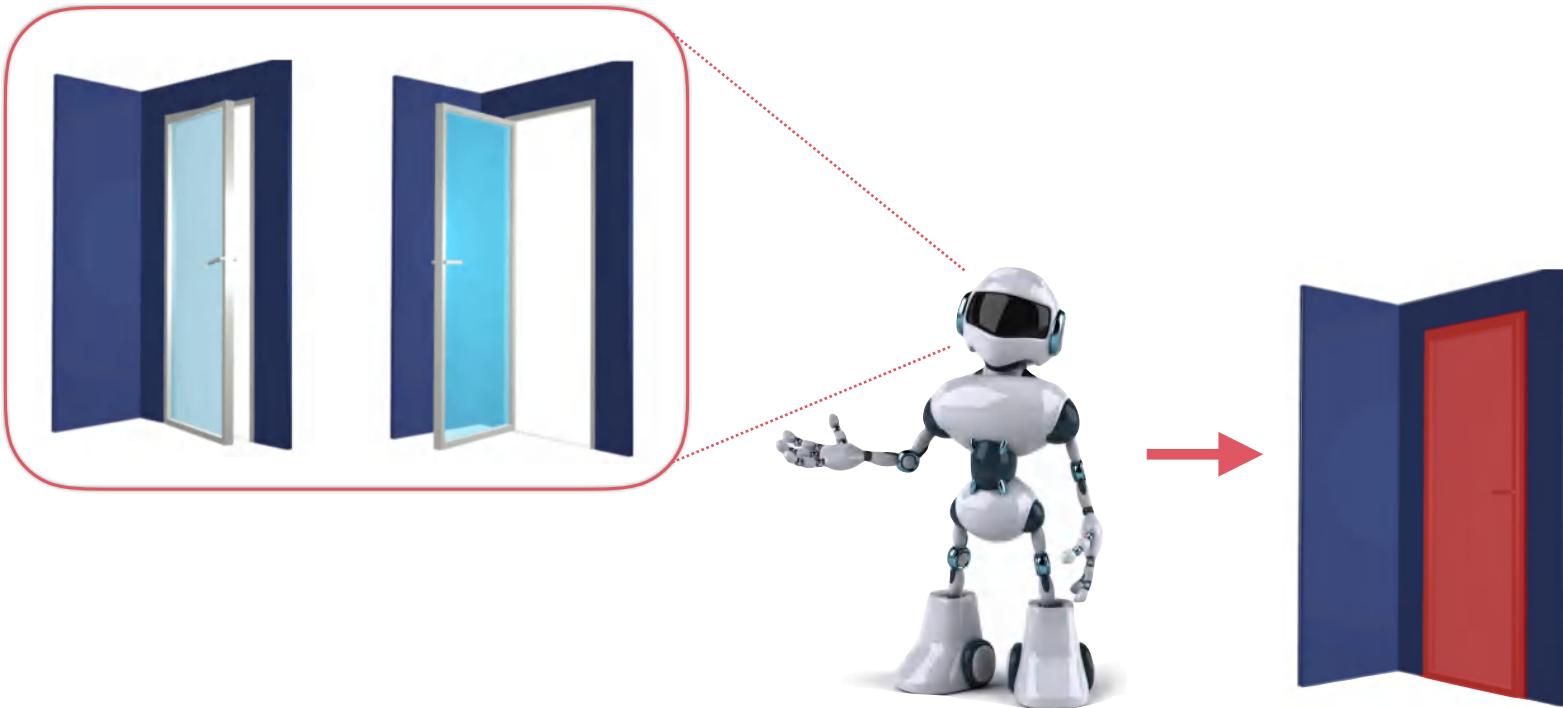
Algorithmically, 3D shapes are:

- easier to be **compared**
- easier to be **related (correspondence)**
- easier to **abstracted**

Task: Few-shot Structure Induction

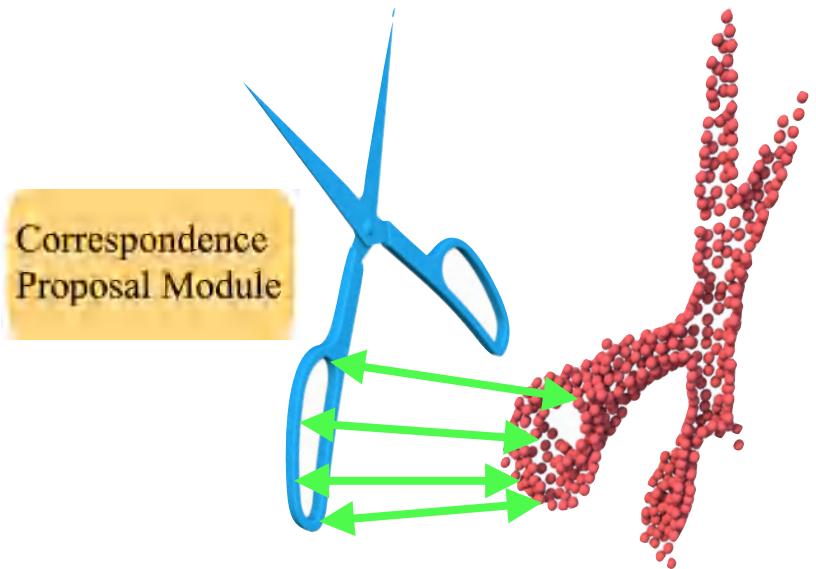


Emergence of Structure by Persistence

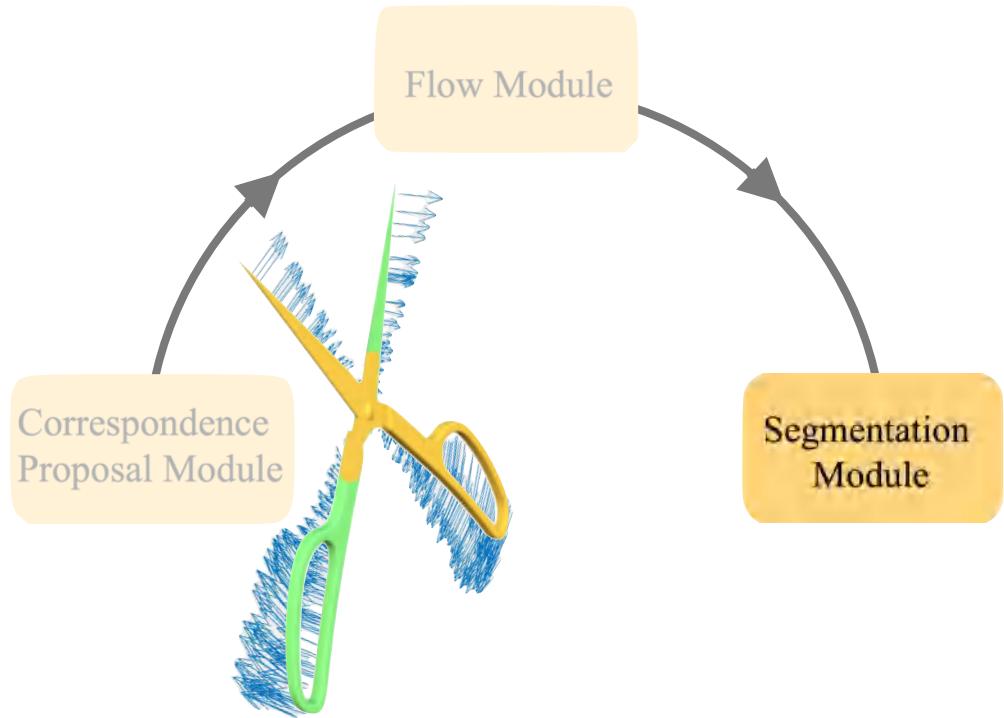


Capture re-occurring units!

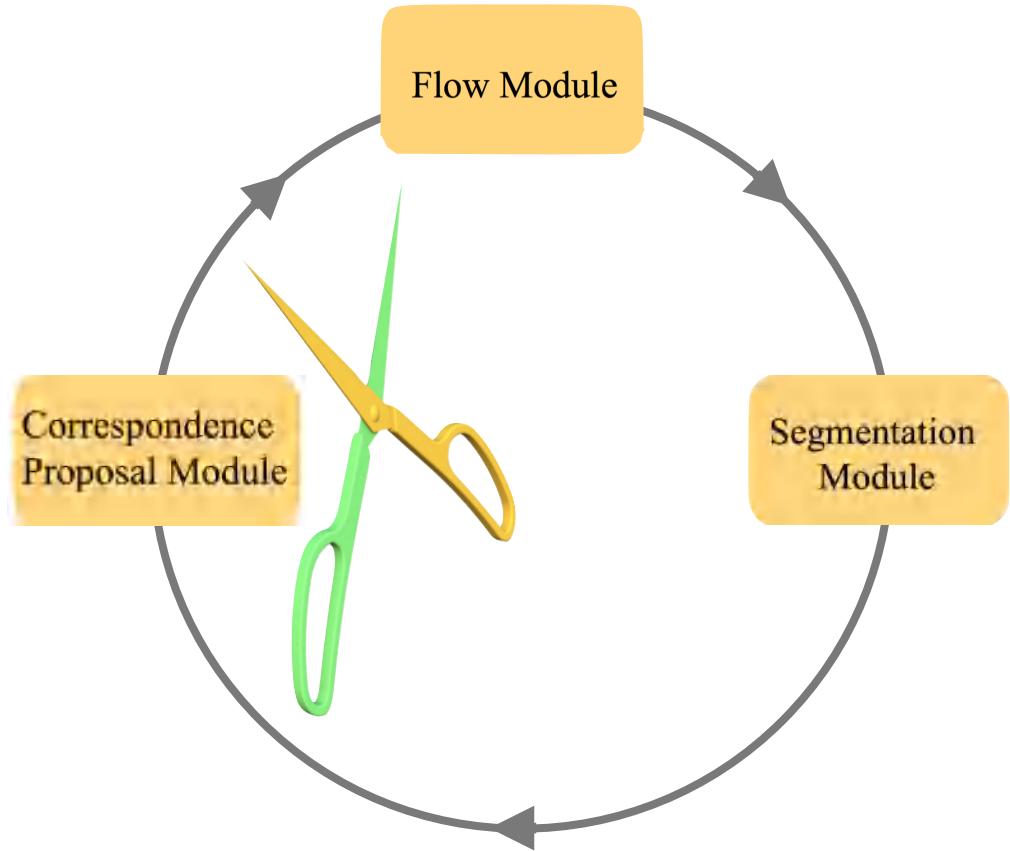
Part Induction by Relating Shapes



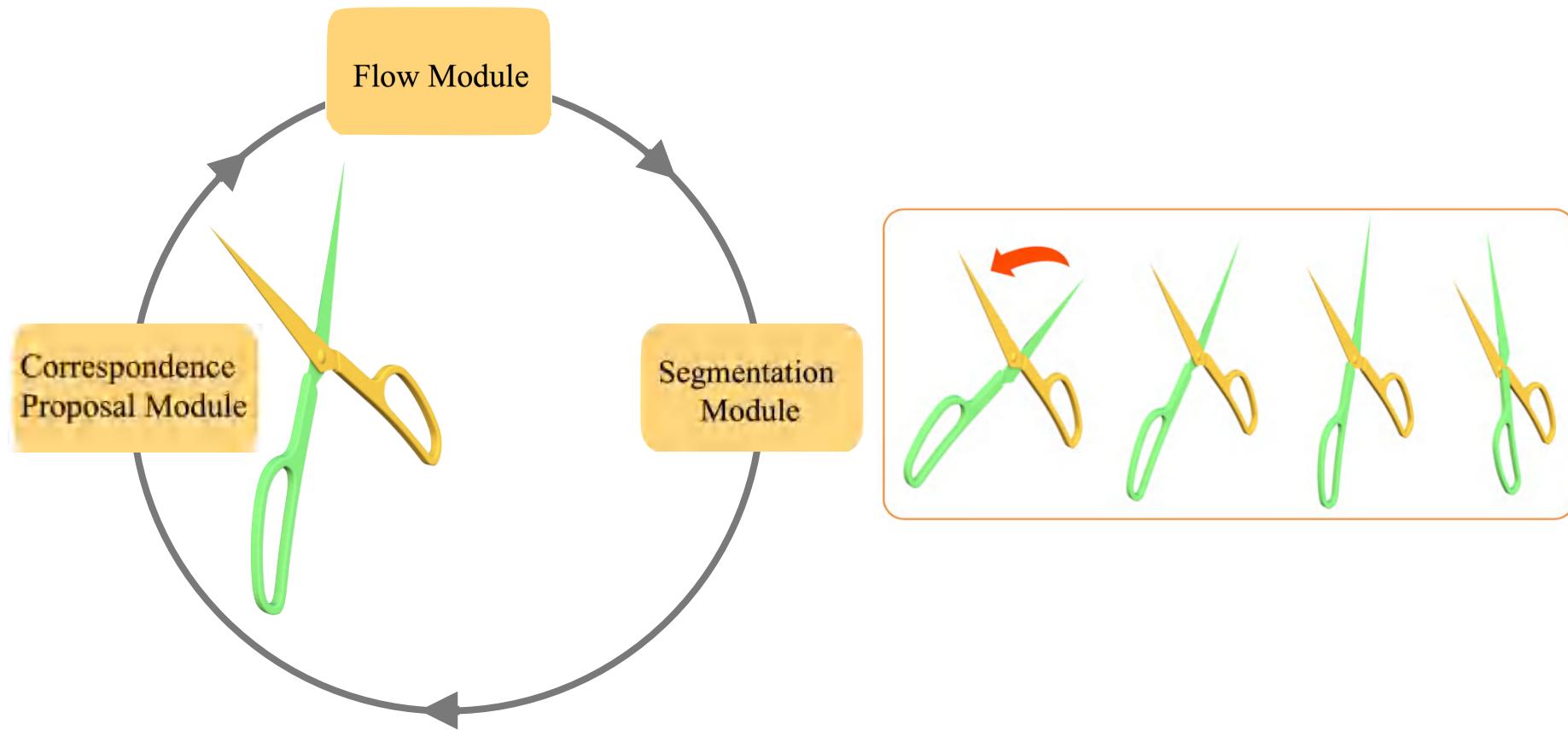
Part Induction by Relating Shapes



Part Induction by Relating Shapes



Mobility Induction

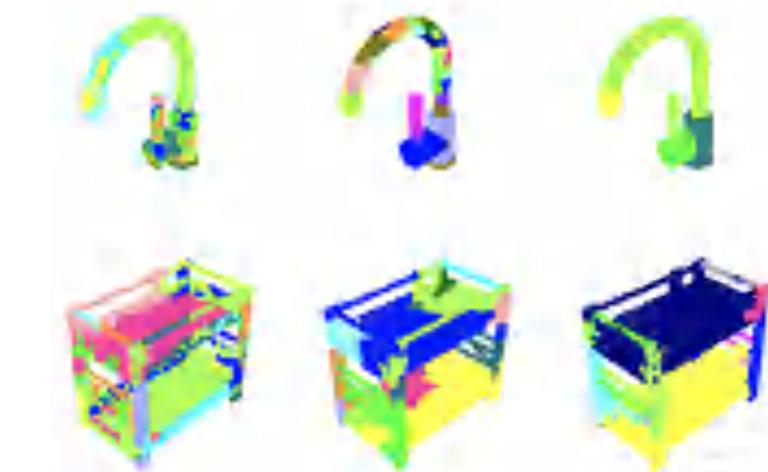


Task: Zero-shot Part Discovery

Train set



Test set



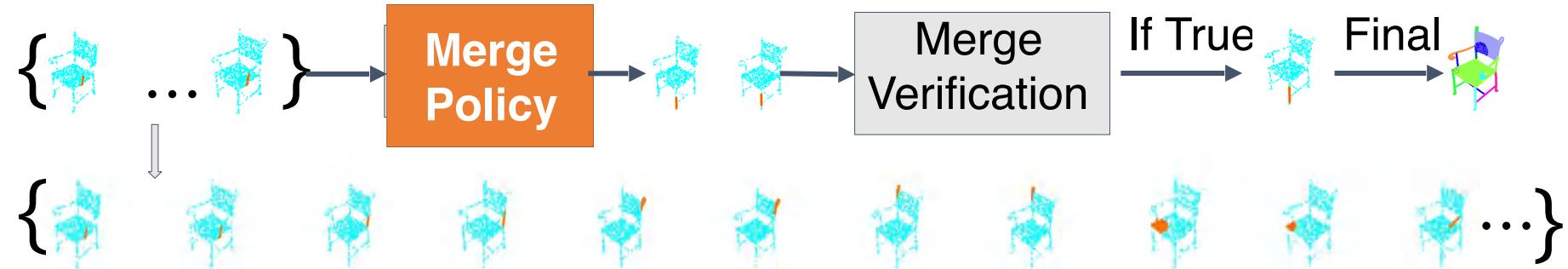
SOTA of
Deep Learning Classics
(PartNet)

SOTA of
Classics
(WCSeg)

Ours

Learning to Group

Sub-Part Pool



Slides will be posted on
<http://ai.ucsd.edu/~haosu/>

(Homepage of Prof. Hao Su)