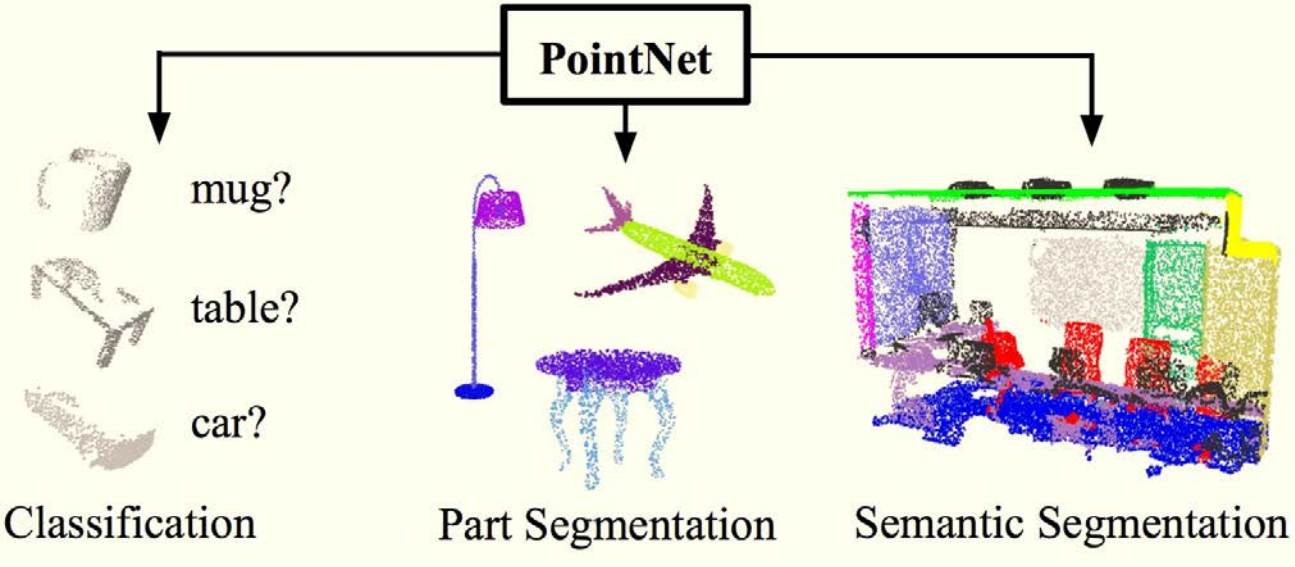
CS233: Geometric and Topological Data Analysis

Deep nets for point clouds 21 May 2018



Lecturer: Charles R. Qi

Agenda

- Deep Nets for Point Cloud Analysis
  - PointNet, PointNet++, Dynamic Graph CNN
  - 3D Object Detection with PointNets
- Deep Nets for Point Cloud Generation

Image Understanding: From feature engineering to learning

Feature Engineering

SIFT
[Lowe, 1999]

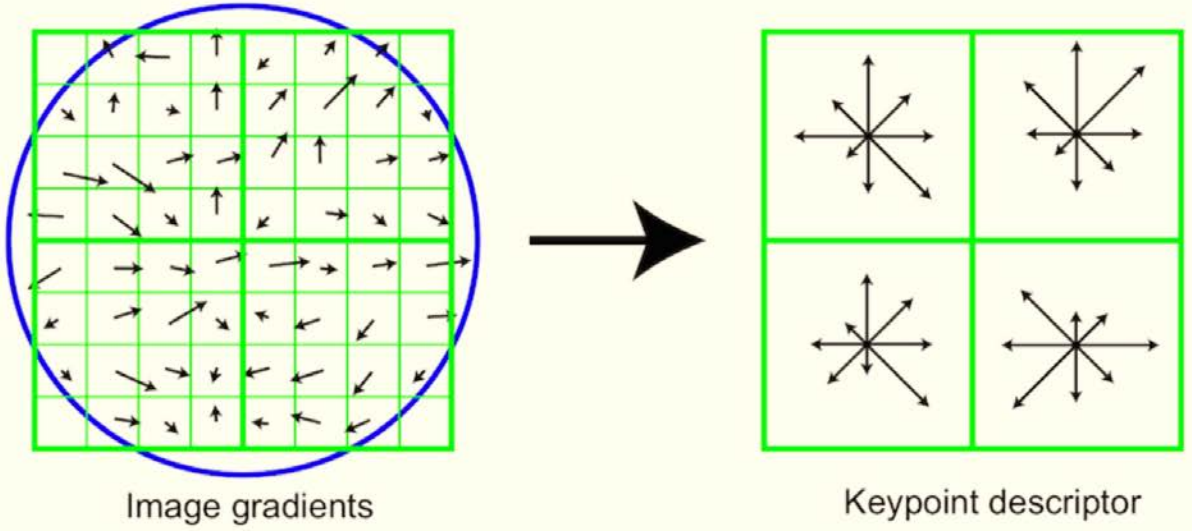
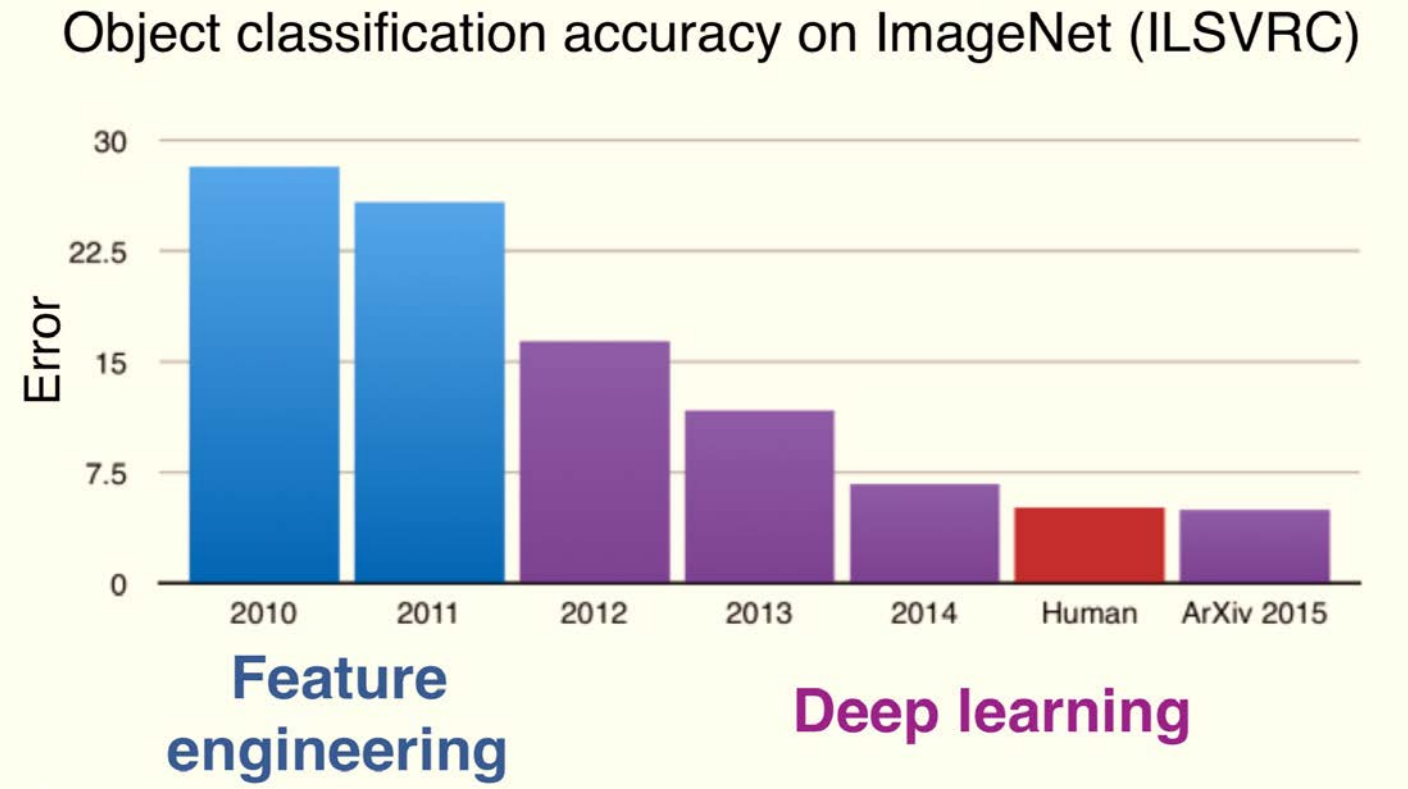


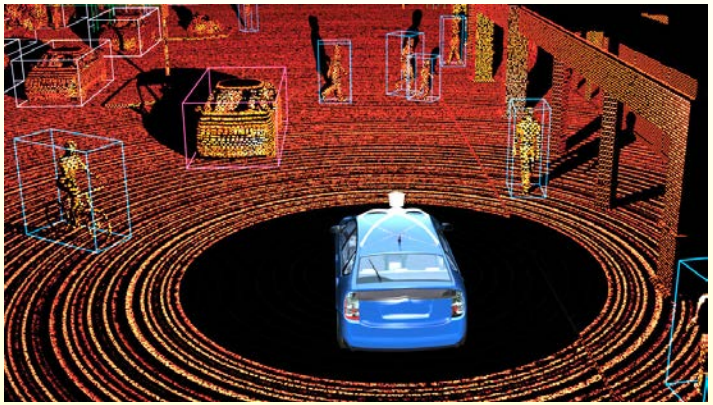
Image Understanding: From feature engineering to learning

Feature Learning



3D Understanding: Emerging applications

Robot Perception

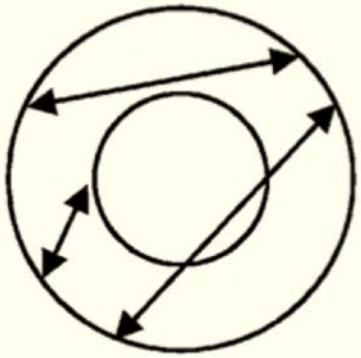


AR/VR

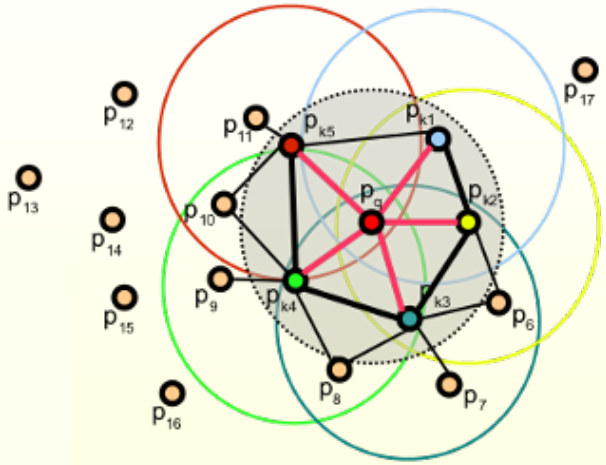
Shape Analysis

Need for 3D Deep Learning!

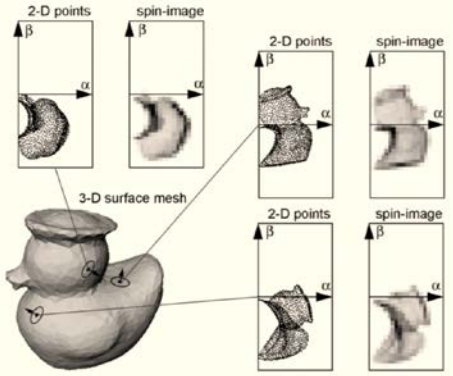
3D Understanding: Prior art



D2
[Osada, 2002]



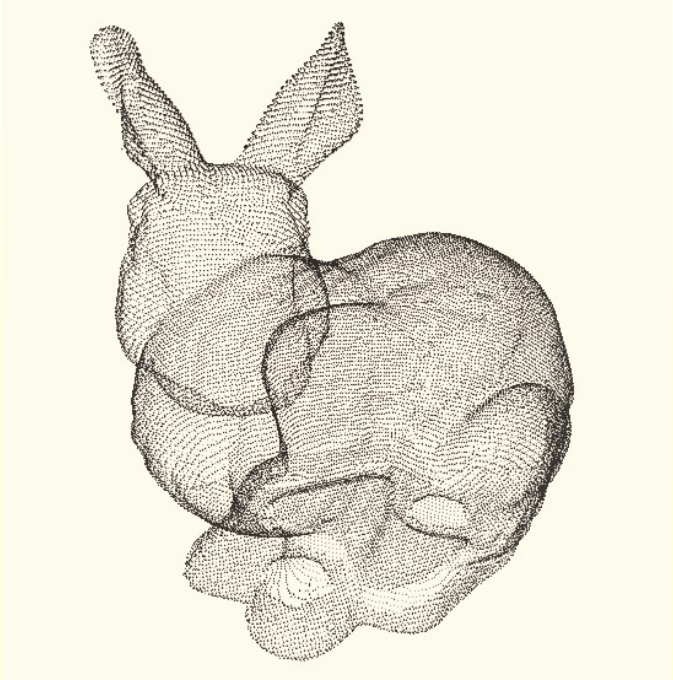
Point Feature Histograms
[Rusu, 2009]



Spin Images
[Johnson, 1999]

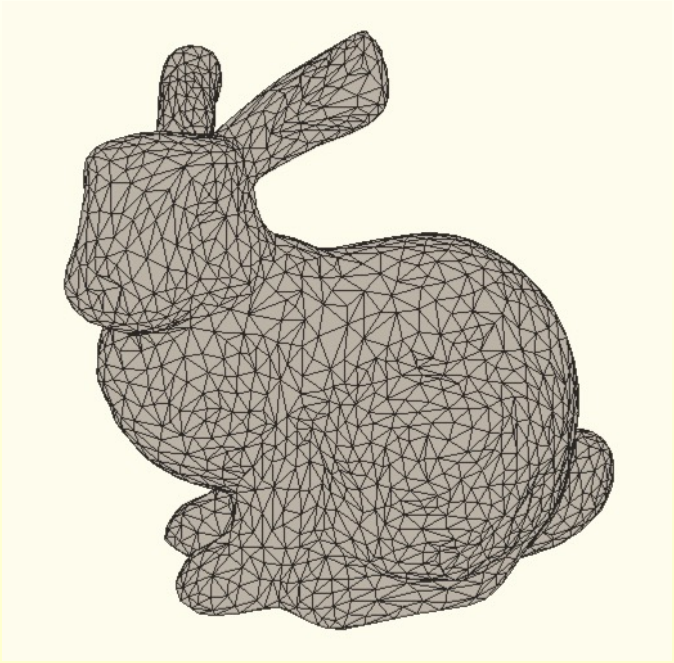
Fundamental challenges for 3D deep learning

Irregularity



Point Cloud

(the most common 3D sensor data)

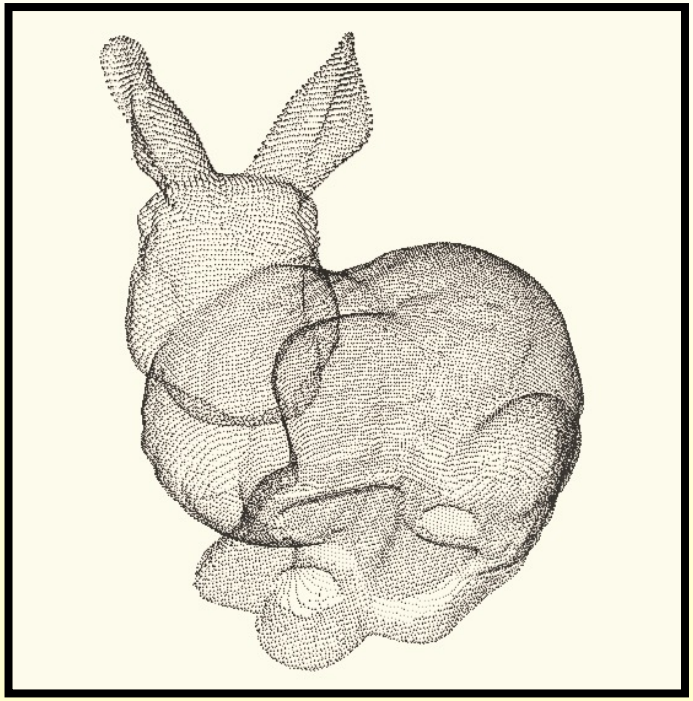


Mesh

(the most common 3D modeling data)

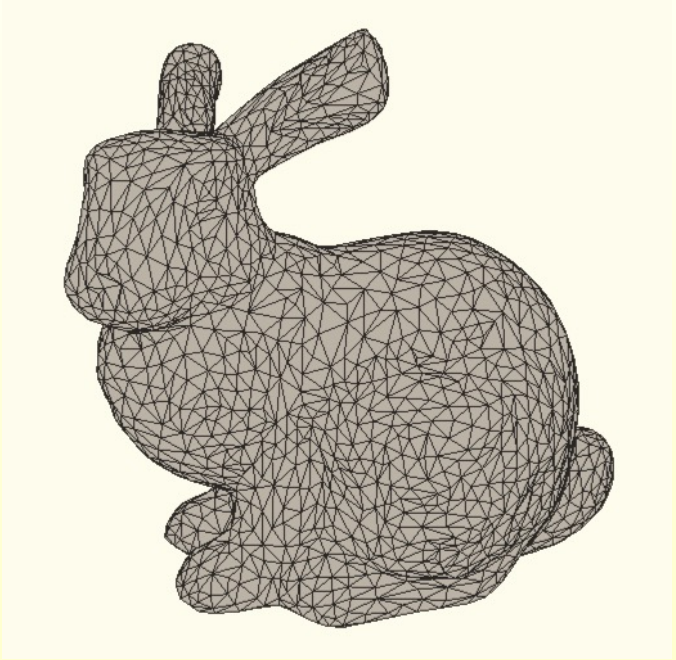
Fundamental challenges for 3D deep learning

Irregularity



Point Cloud

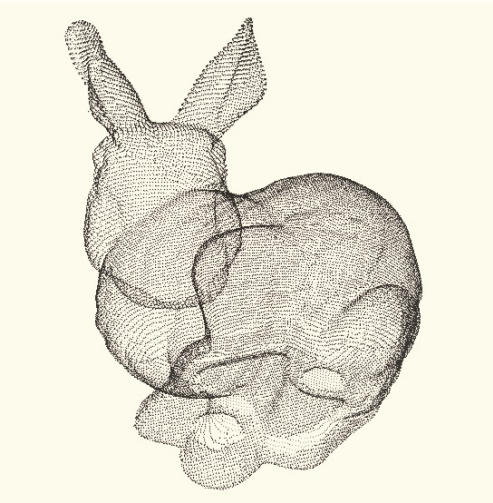
(the most common 3D sensor data)

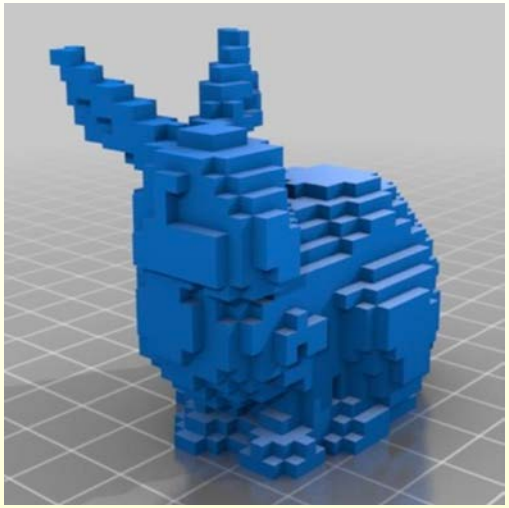


Mesh

(the most common 3D modeling data)

Solution 1: Convert irregular to regular





Point Cloud

Volumetric

High space/time complexity -- 3D convolution $O(N^3)$!

Information loss in voxelization!

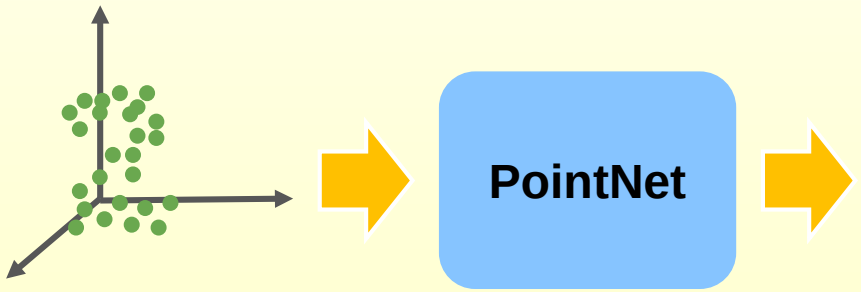
Solution 1: Convert irregular to regular

or.. point clouds can be converted to images or hand-crafted feature vectors before it is fed into a deep neural network for feature learning.

Conversion	Deep Net
Voxelization	3D CNN
Projection/Rendering	2D CNN
Feature extraction	Fully Connected

Solution 2: Directly learning on point clouds

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



Object Classification

Part Segmentation

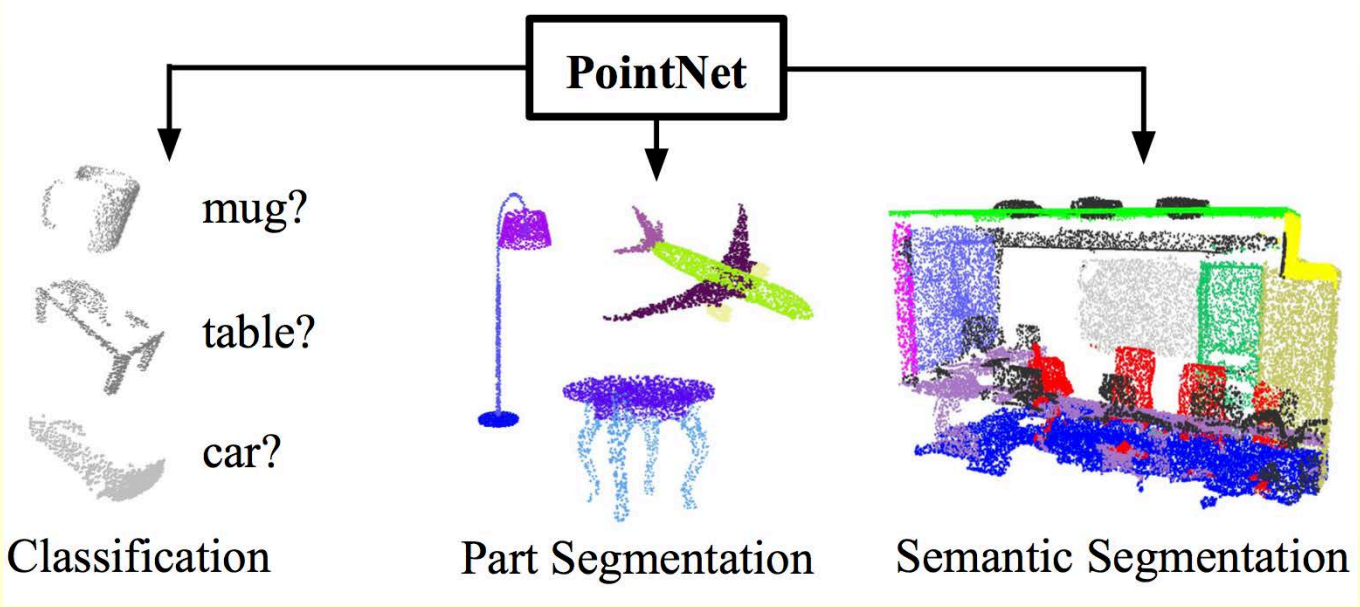
Semantic Scene Parsing

...

PointNet

Problem Formulation

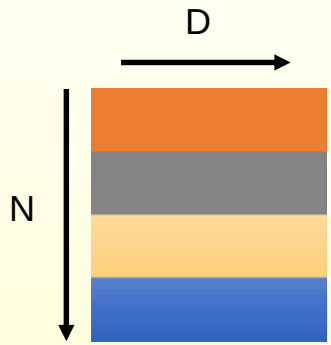
Input: **point clouds (2D array)**



Output: **class scores** **per point class scores**

Desired properties of a deep neural network on point clouds

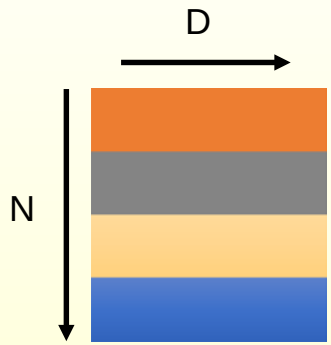
Point cloud: N **orderless** points, each represented by a D dim vector



2D array representation of a point cloud

Desired properties of a deep neural network on point clouds

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

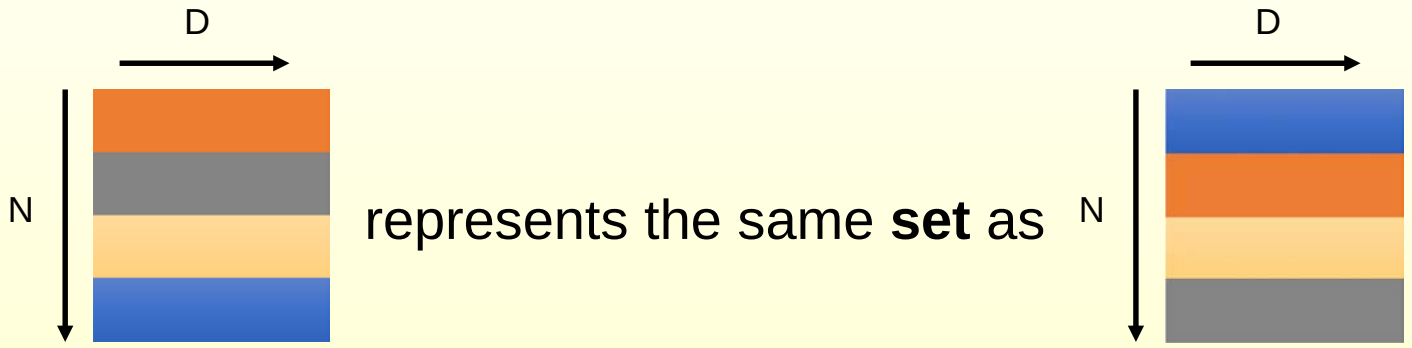


2D array representation of a point cloud

- **Permutation invariance**
- **Transformation invariance**

Unordered Points

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



Model needs to be invariant to N! permutations

Permutation Invariance: Symmetric Function

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

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:

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

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

…

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:

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

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

…

How can we construct a family of symmetric functions by neural networks?

Construct symmetric function family

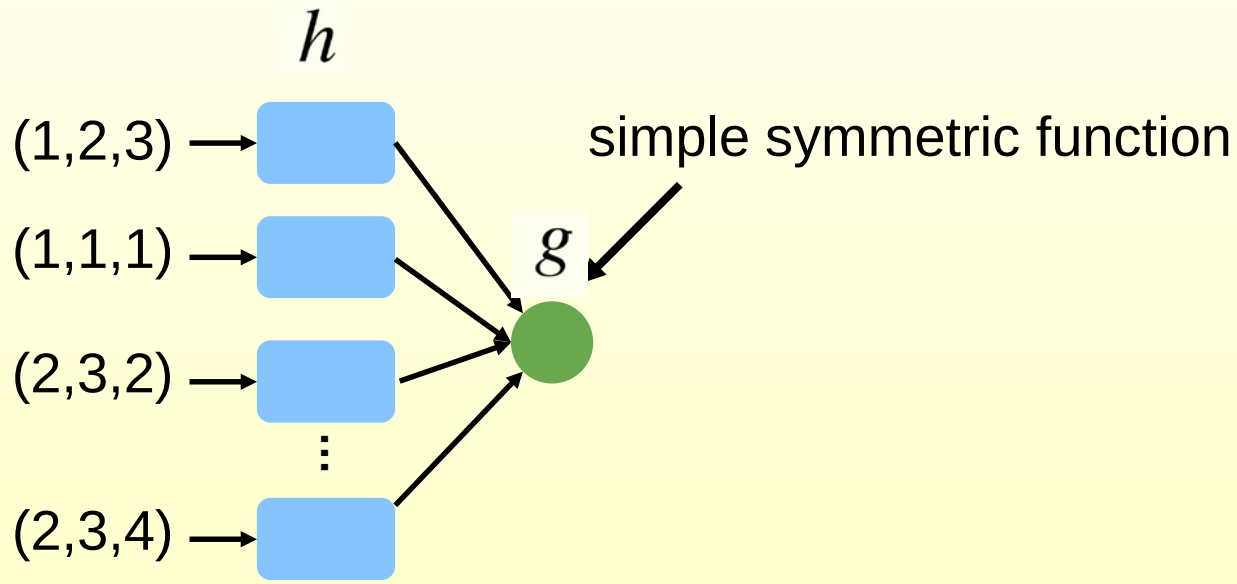
Observe:

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

Construct symmetric function family

Observe:

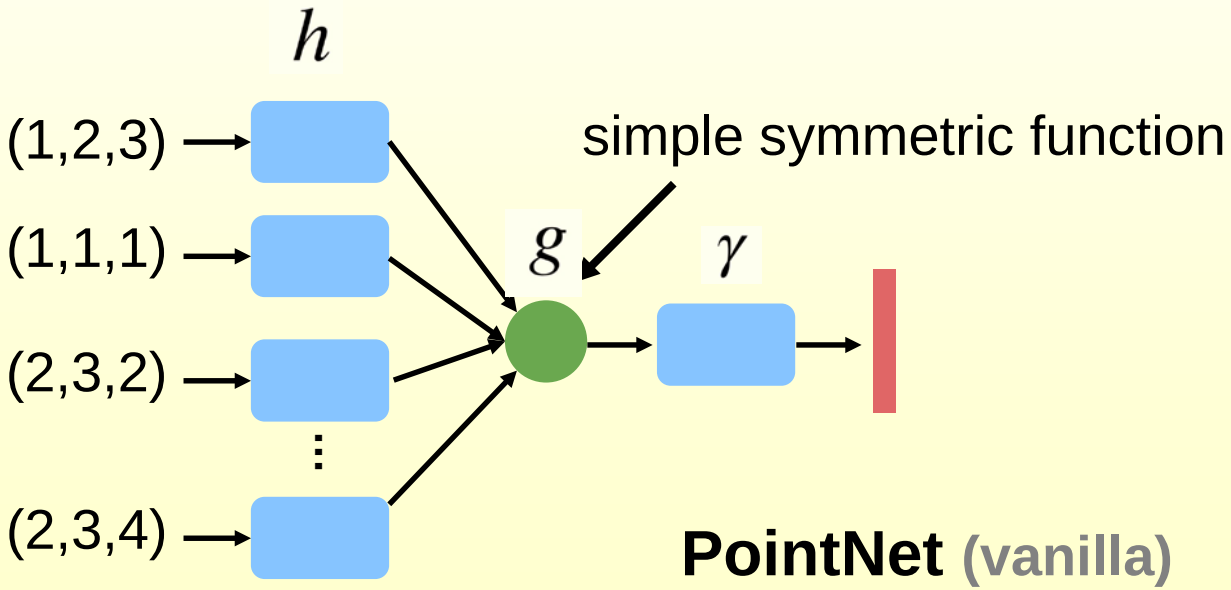
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Construct symmetric function family

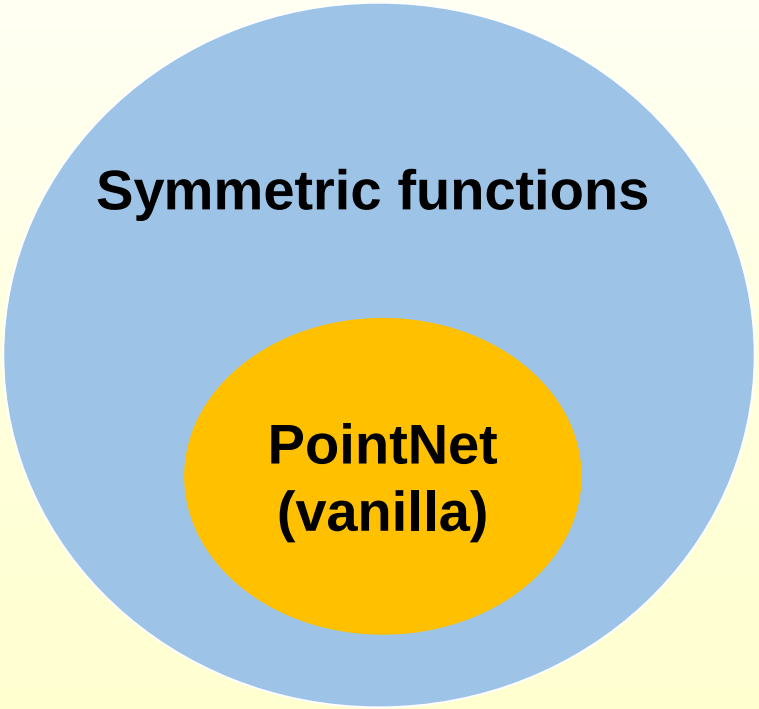
Observe:

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



PointNet (vanilla)

Q: What symmetric functions can be constructed by PointNet?



A: Universal approximation to continuous symmetric functions

Theorem:

A Hausdorff continuous symmetric function $f : 2^{\mathcal{X}} \to \mathbb{R}$ can be arbitrarily approximated by PointNet.

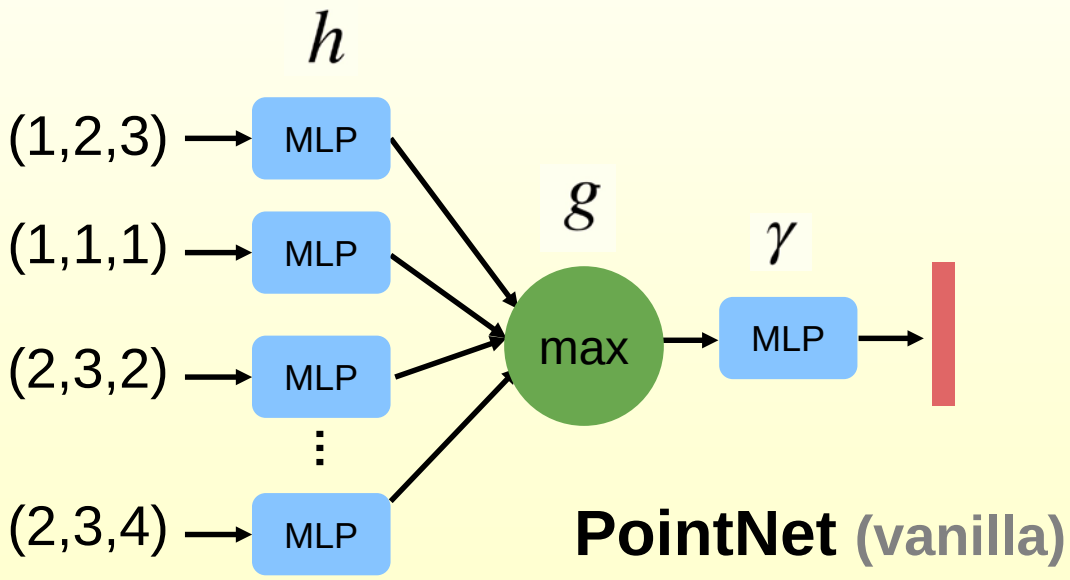
$$\left| f(S) - \gamma \left(\underset{x_i \in S}{\mathrm{MAX}} \{ h(x_i) \} \right) \right| < \epsilon$$

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

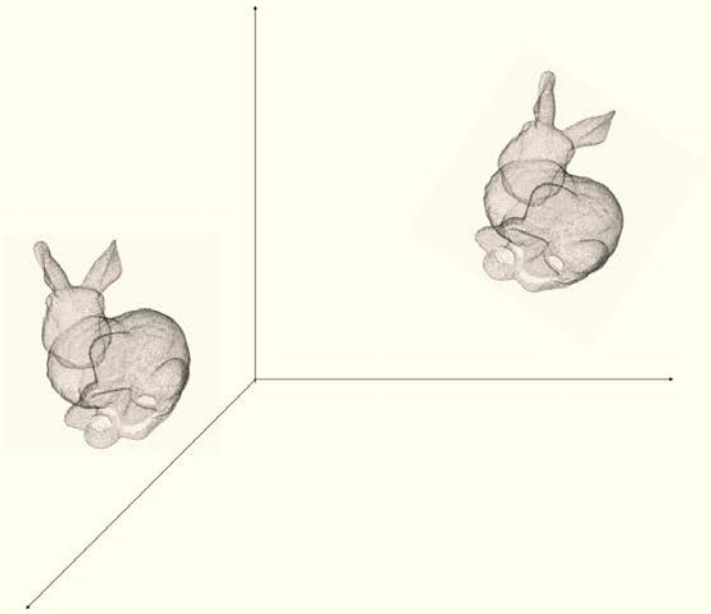
PointNet (vanilla)

Basic PointNet Architecture

Empirically, we can use **multi-layer perceptron (MLP)** and **max pooling**:



Transformation invariance is desirable

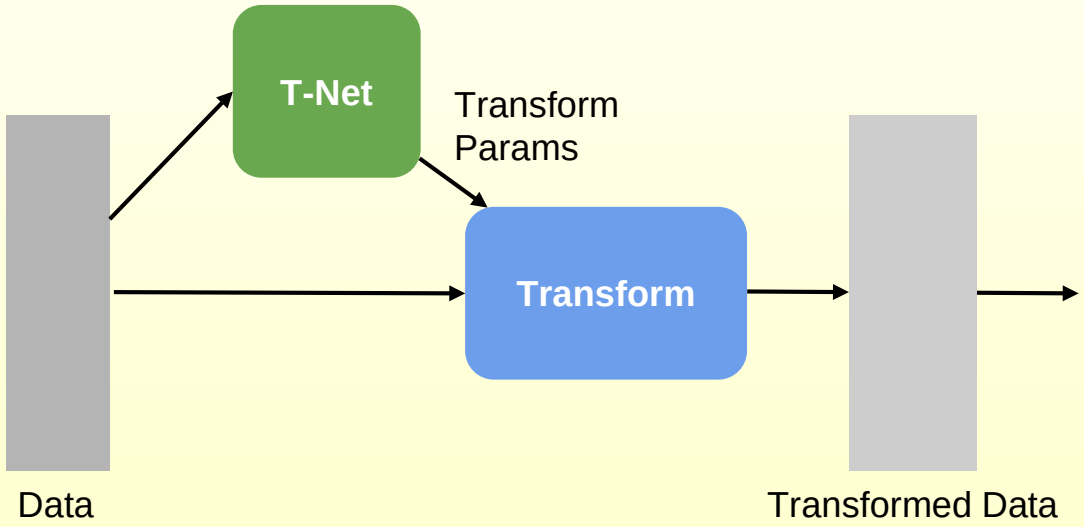


Let S be a shape. Then $f(T \cdot S) = f(S)$

f : classifier, T : transformation matrix

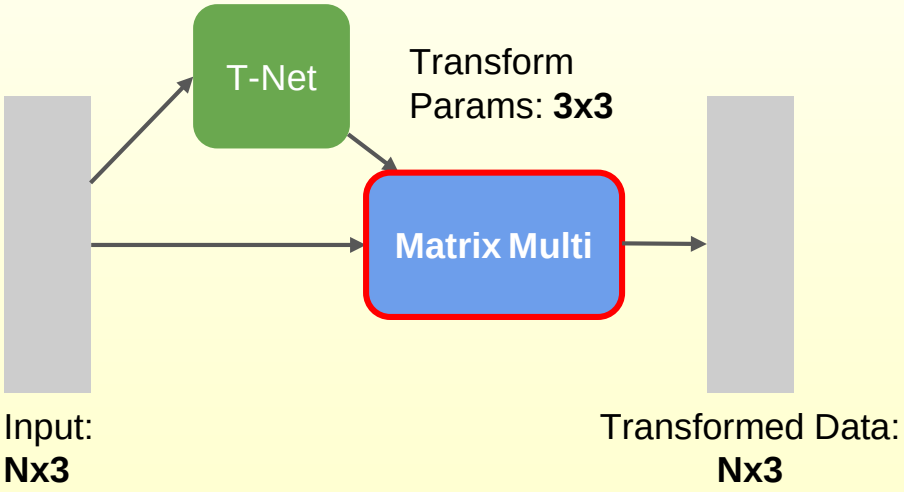
Using transformer networks

Idea: Data dependent transformation for automatic alignment



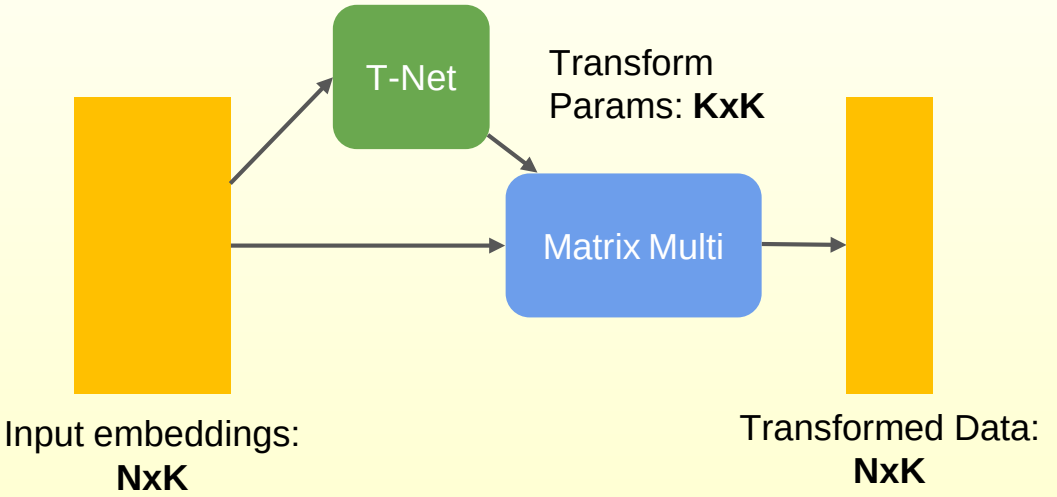
Input and Embedding Space Transformations

(1) Spatial transform. The transformation is just matrix multiplication (compared to bilinear/trilinear sampling in grids)!



Input and Embedding Space Transformations

(2) Embedding space transform.

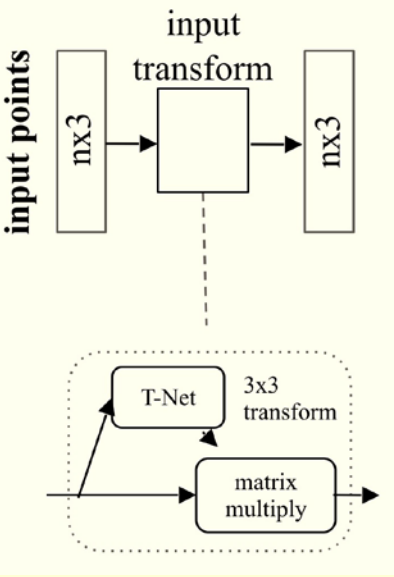


Regularization:
Transform matrix KxK close to orthogonal:

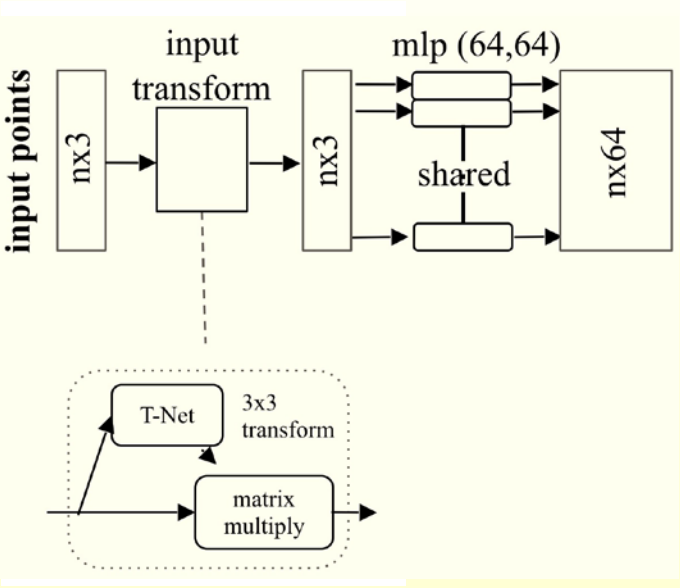
$$L_{reg} = \|I - AA^T\|_F^2$$

PointNet Classification Network

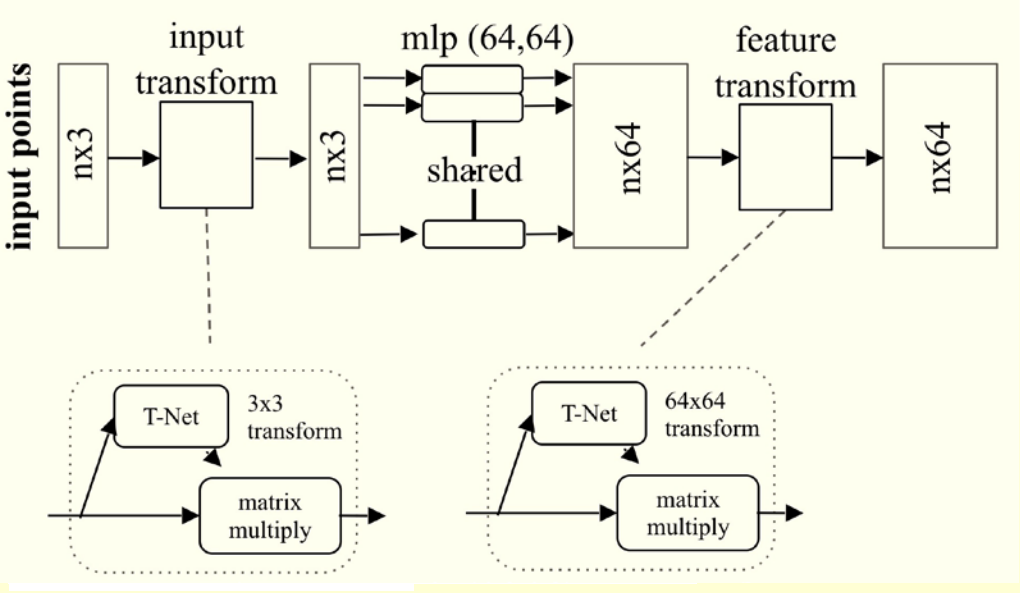
PointNet Classification Network



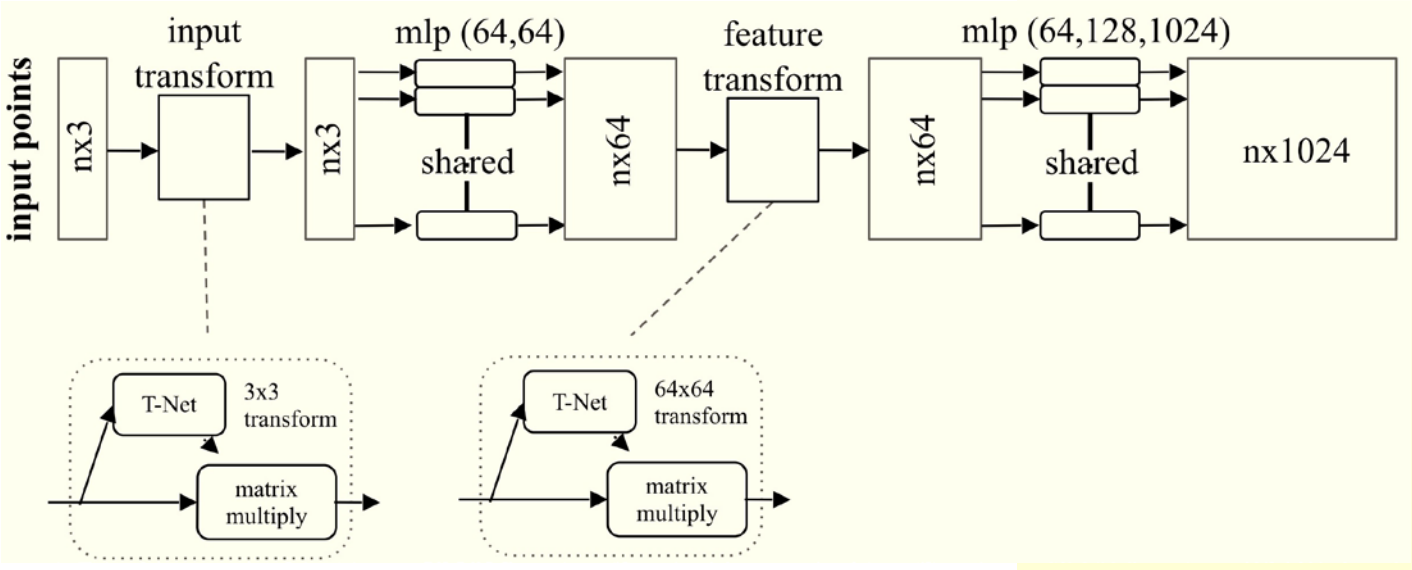
PointNet Classification Network



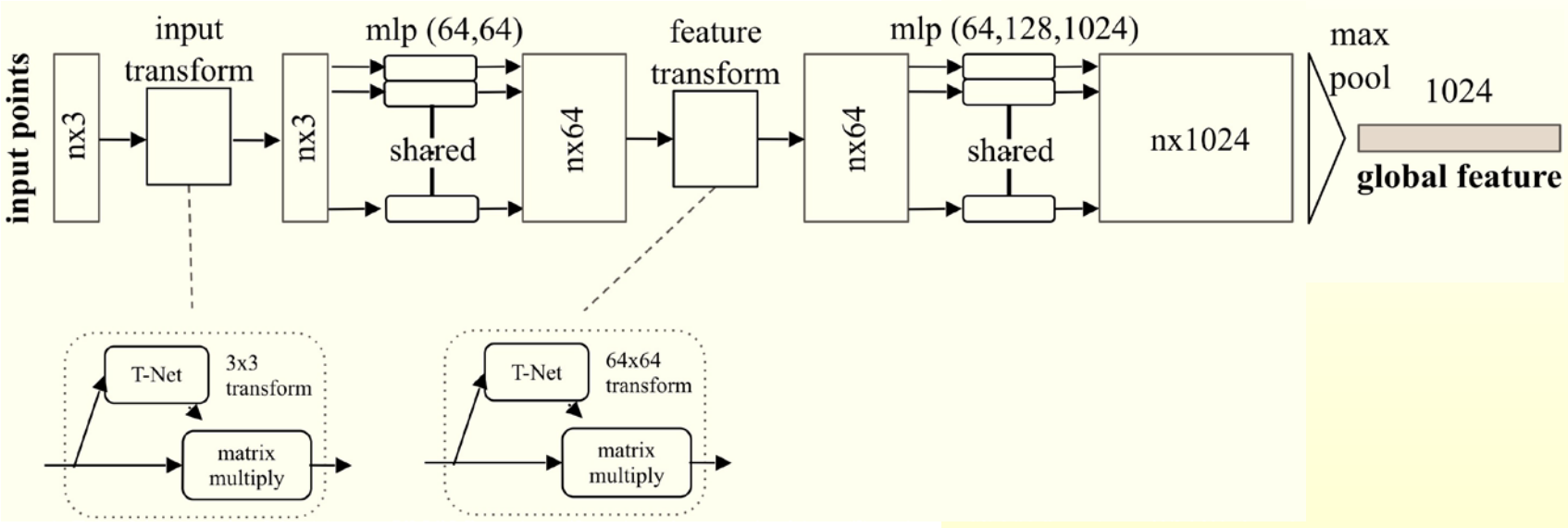
PointNet Classification Network



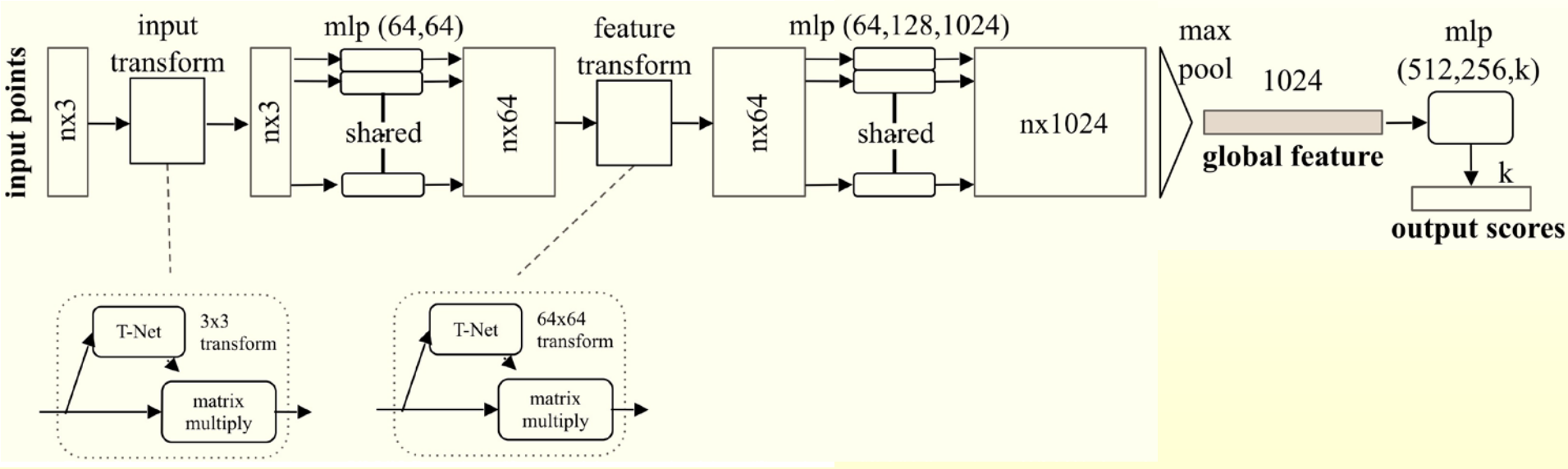
PointNet Classification Network



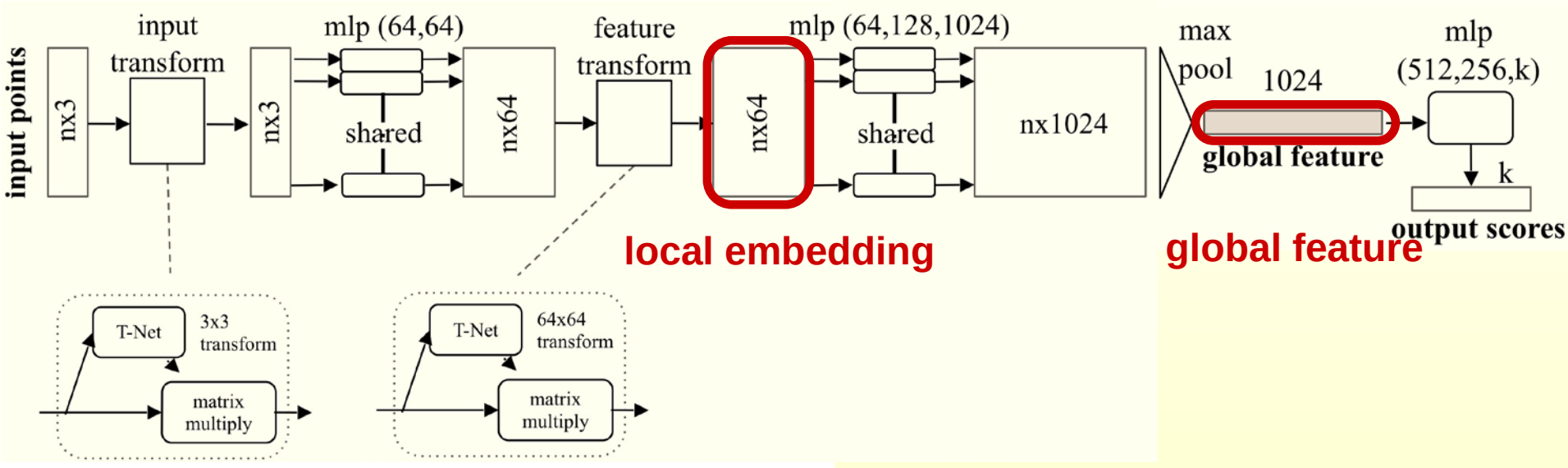
PointNet Classification Network



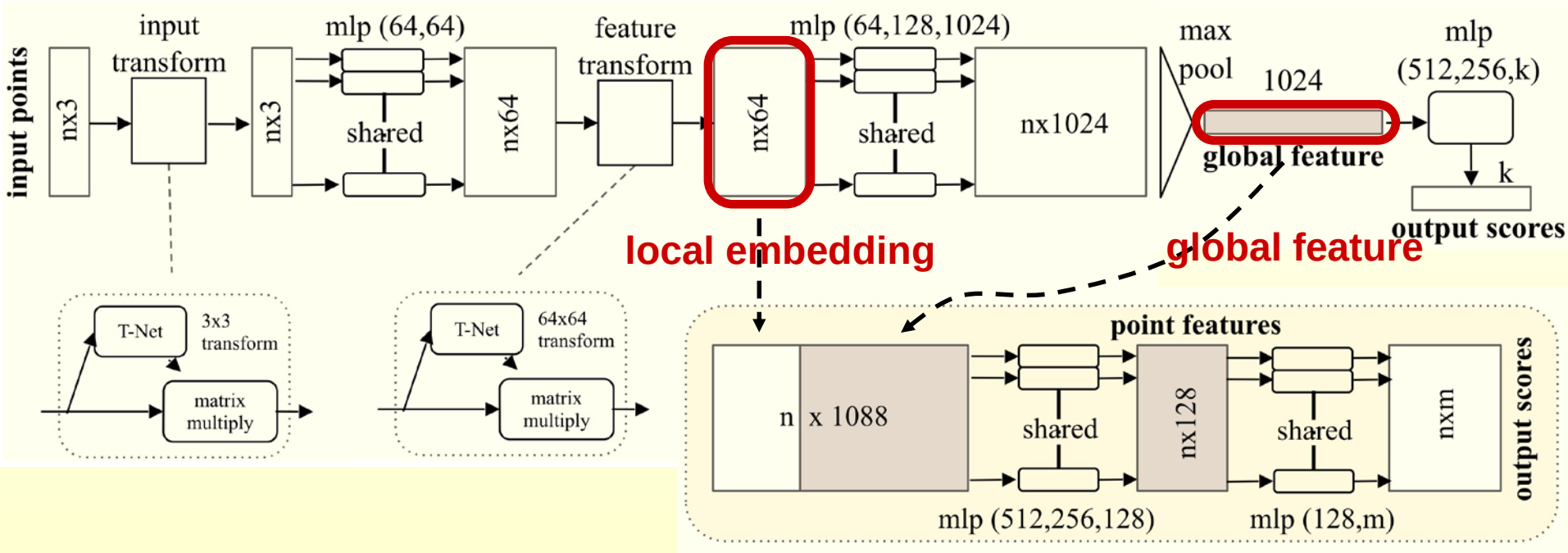
PointNet Classification Network



Extension to PointNet Segmentation Network



Extension to PointNet Segmentation Network



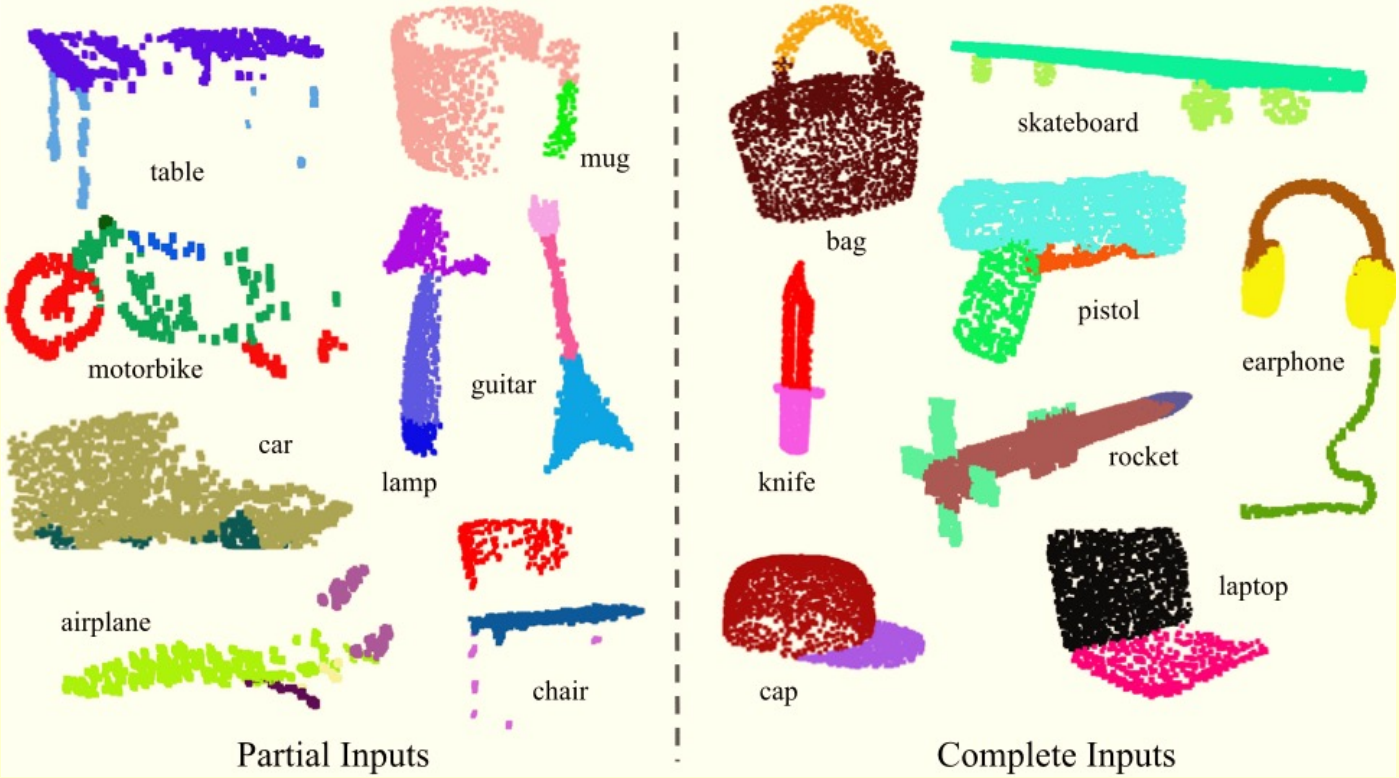
Results on Object Classification

	input	#views	accuracy avg. class	accuracy overall
SPH [12]	mesh	-	68.2	
3DShapeNets [29]	volume	1	77.3	84.7
VoxNet [18]	volume	12	83.0	85.9
Subvolume [19]	volume	20	86.0	**89.2**
LFD [29]	image	10	75.5	-
MVCNN [24]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

3D CNNs

dataset: ModelNet40; metric: 40-class classification accuracy (%)

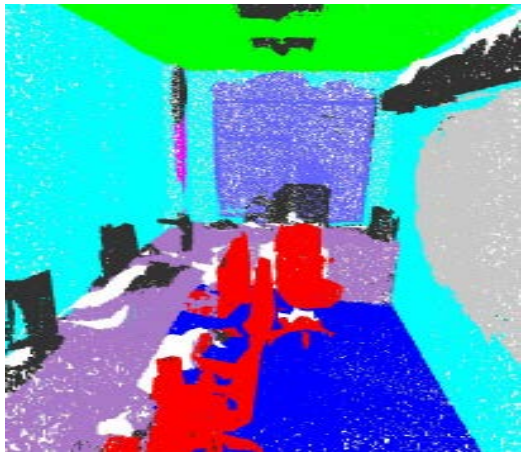
Results on Object Part Segmentation

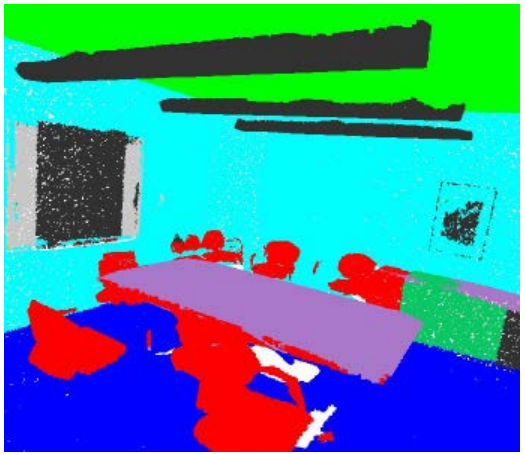


Results on Semantic Scene Parsing

Input

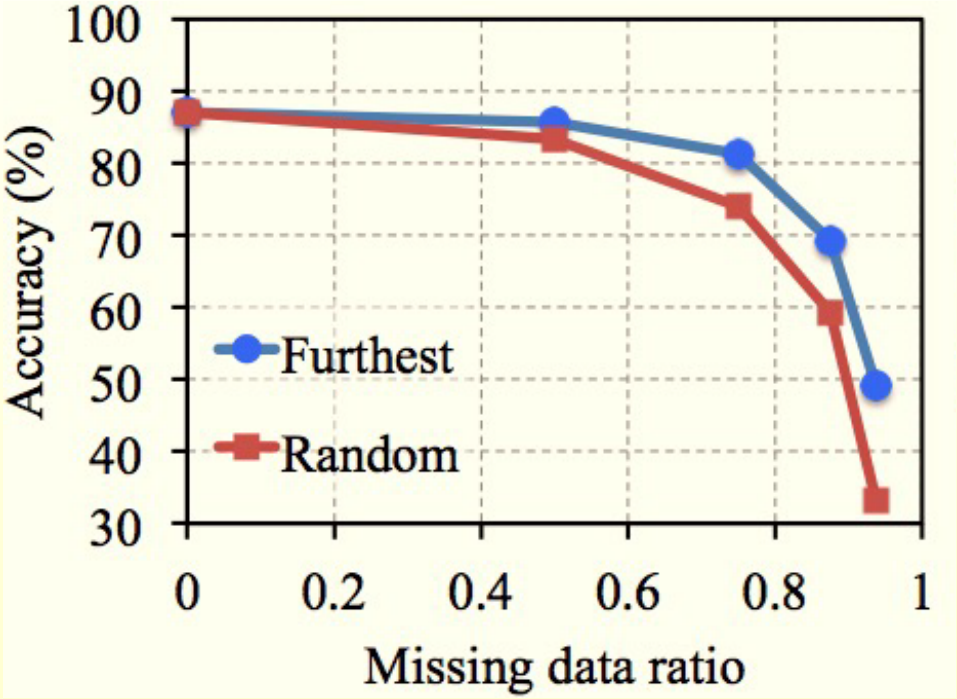
Output





dataset: Stanford 2D-3D-S (Matterport scans)

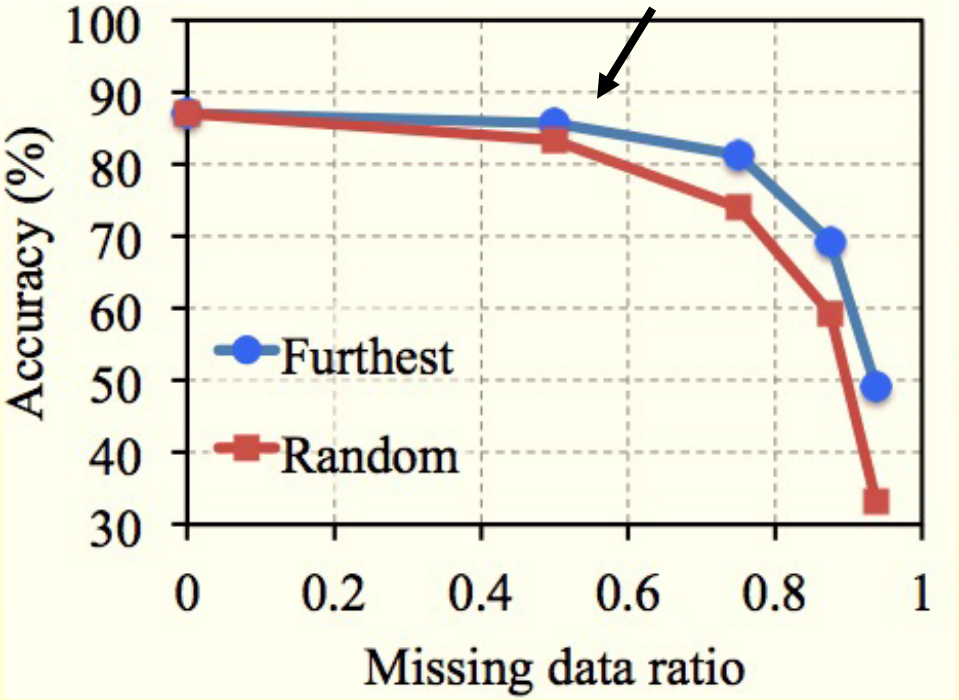
Robustness to Data Corruption



dataset: ModelNet40; metric: 40-class classification accuracy (%)

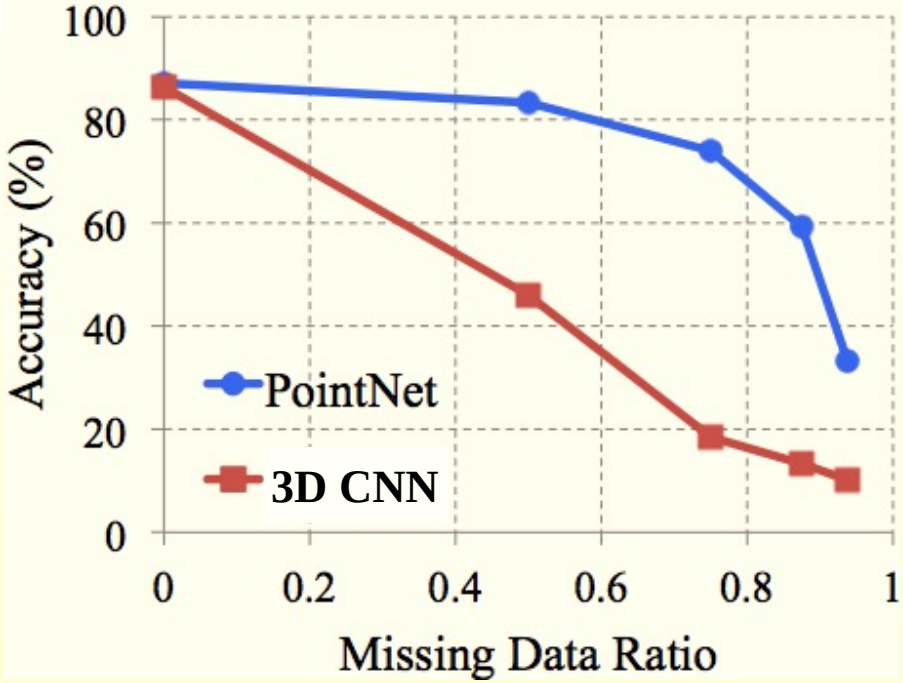
Robustness to Data Corruption

Less than 2% accuracy drop with 50% missing data



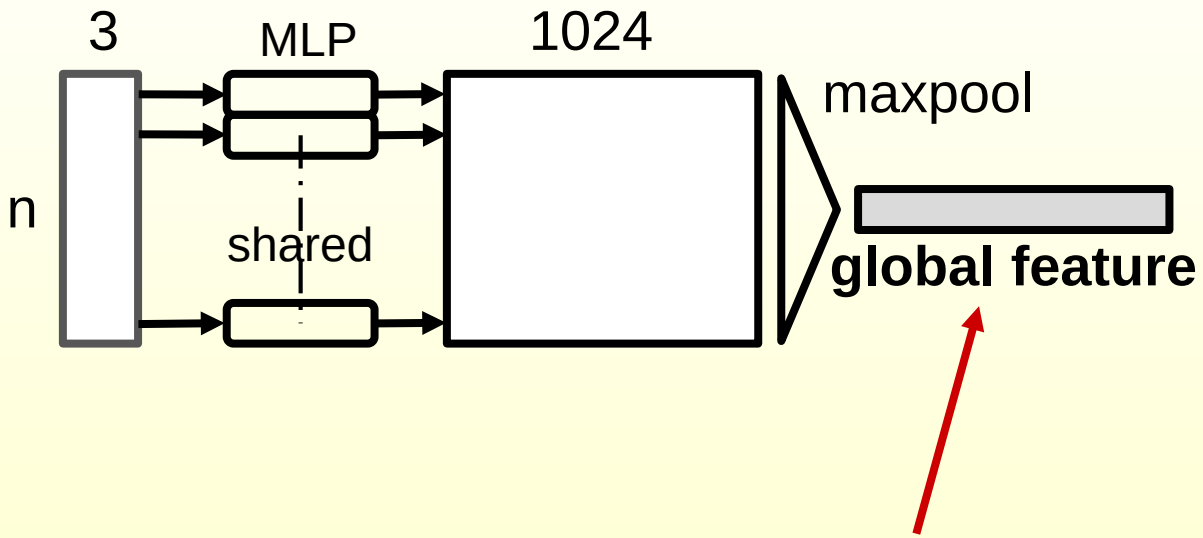
dataset: ModelNet40; metric: 40-class classification accuracy (%)

Robustness to Data Corruption



Why is PointNet so robust to missing data?

Visualizing Global Point Cloud Features



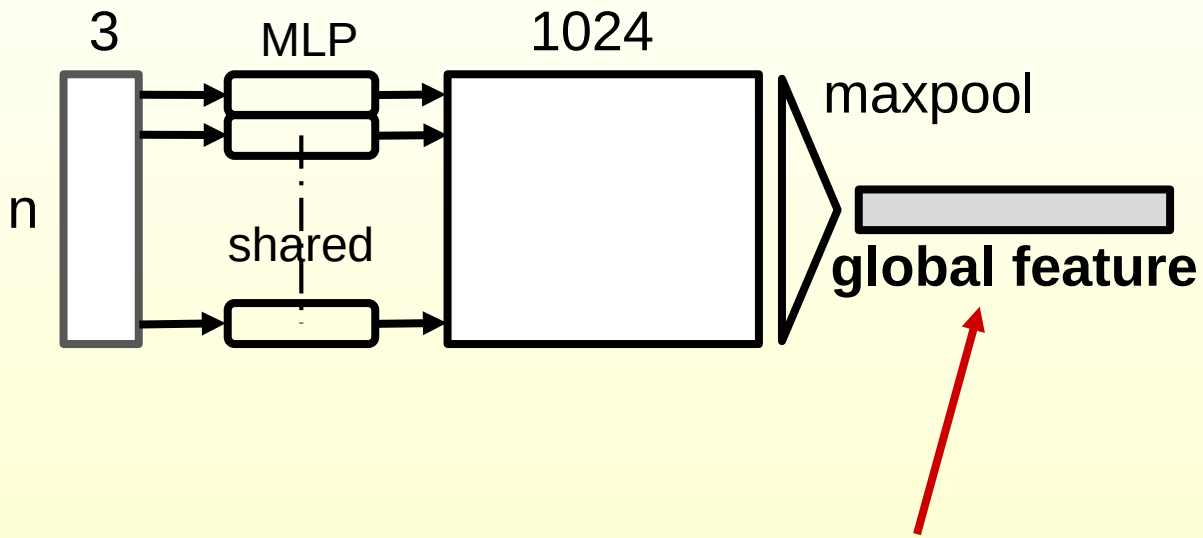
<u>Which input points</u> are contributing to the global feature?
(critical points)

Visualizing Global Point Cloud Features

Original Shape:

Critical Point Sets:

Visualizing Global Point Cloud Features



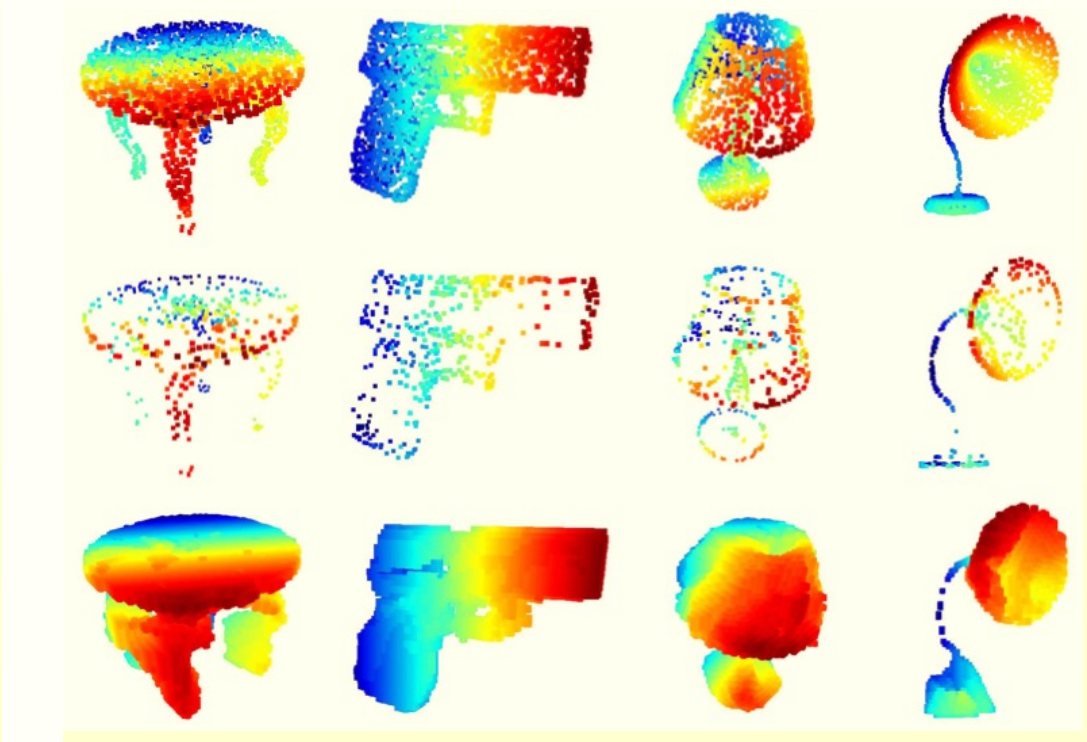
<u>Which points</u> won't affect the global feature?

Visualizing Global Point Cloud Features

Original Shape:

Critical Point Set:

Upper bound set:

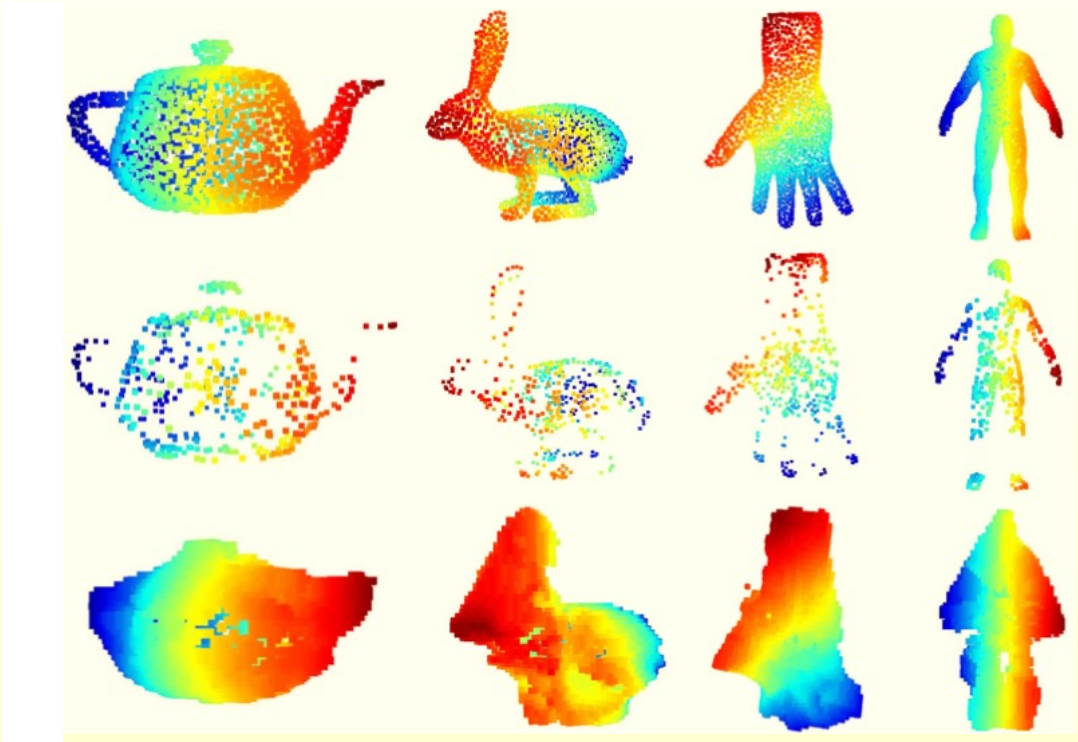


Visualizing Global Point Cloud Features (out-of-sample)

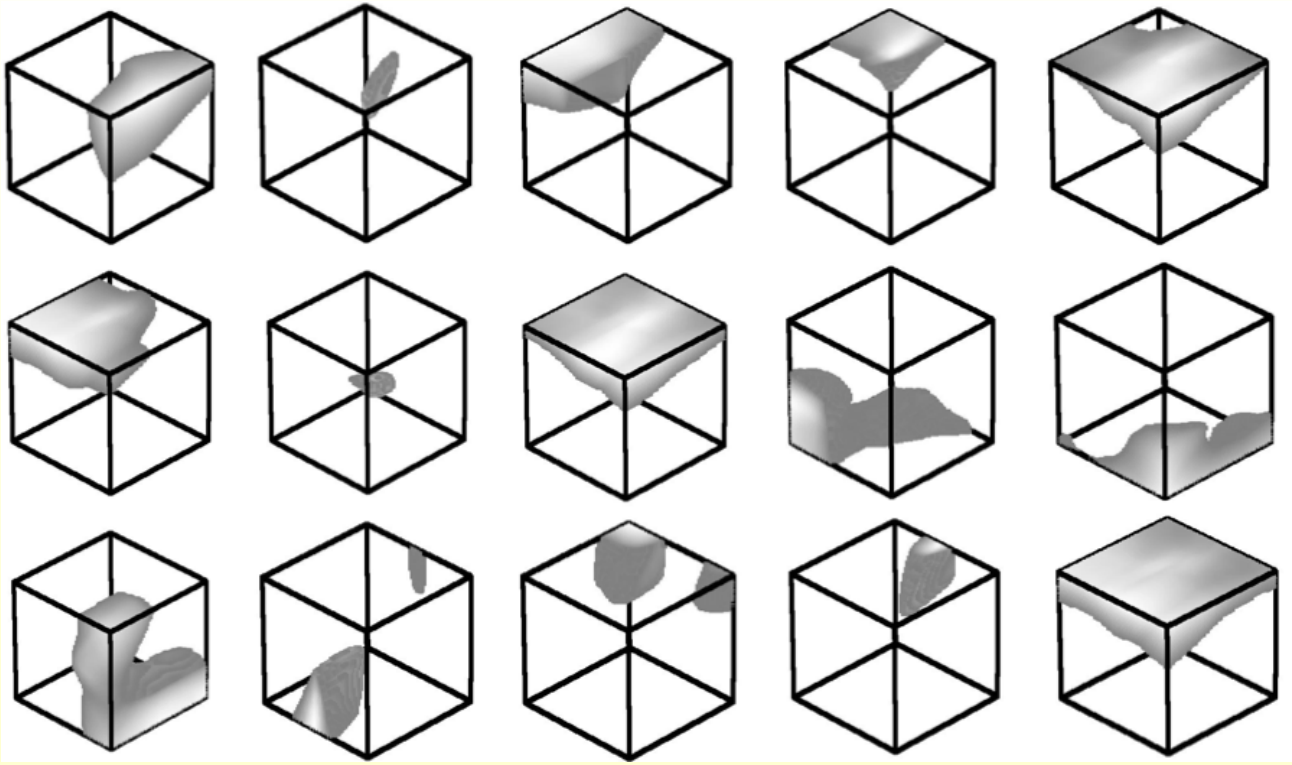
Original Shape:

Critical Point Set:

Upper bound Set:



Visualizing Point Functions



Efficiency of PointNet

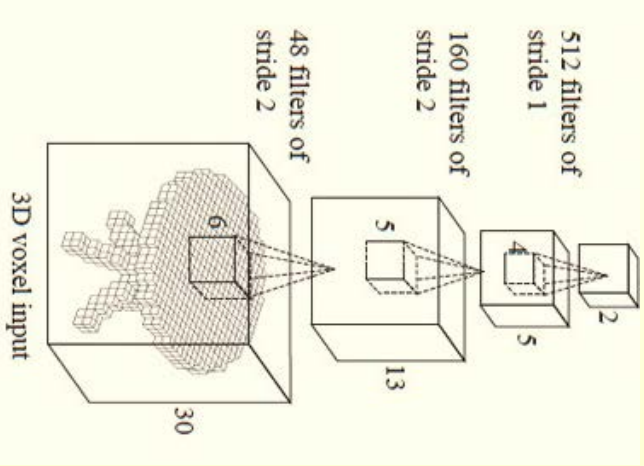
	#params	FLOPs/sample
PointNet (vanilla)	0.8M	148M
PointNet	3.5M	440M
Subvolume [16]	16.6M	3633M
MVCNN [20]	60.0M	62057M

A promising architecture for portable devices!

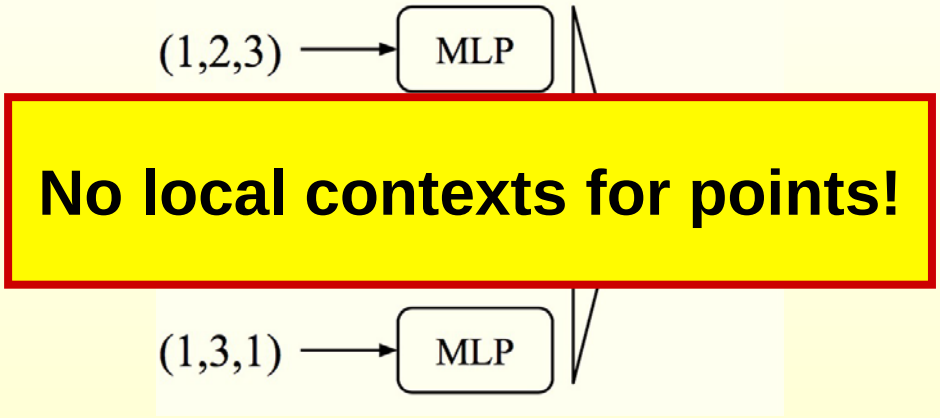
Limitations of PointNet

Hierarchical feature learning
multiple levels of abstraction

Global feature learning
Either one point or all points



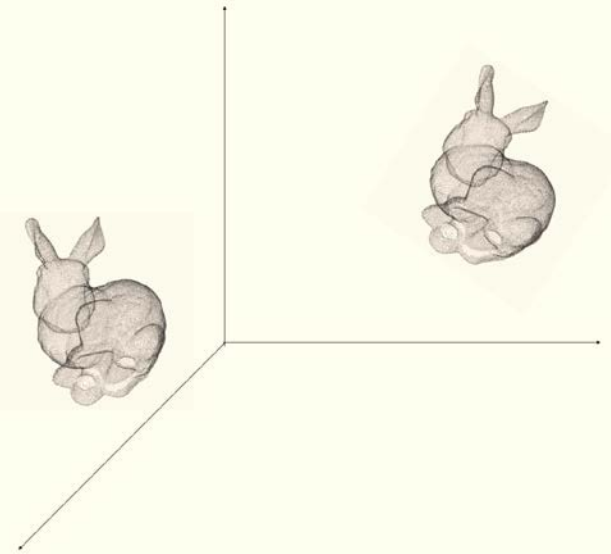
3D CNN (Wu et al.)

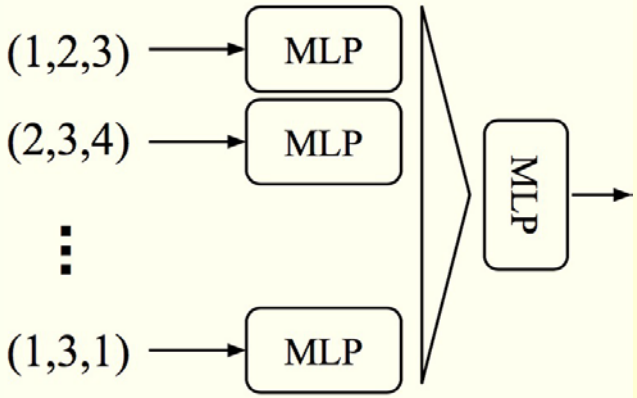


PointNet (vanilla) (Qi et al.)

Limitations of PointNets

Desired property of deep neural network on point clouds: **Transformation Invariance**





PointNet is NOT translation invariant!

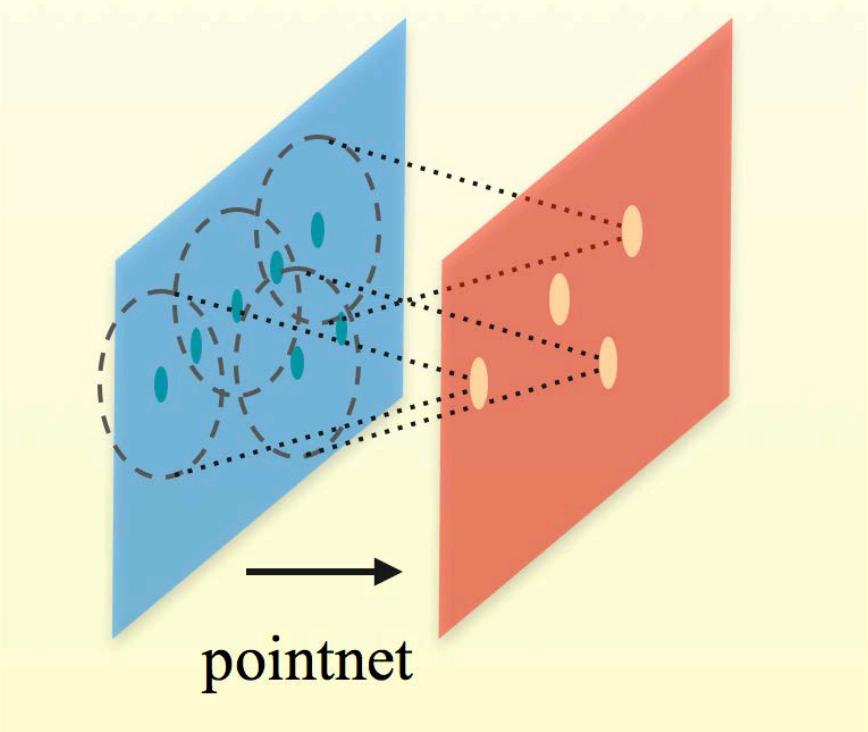
PointNet++

PointNet++

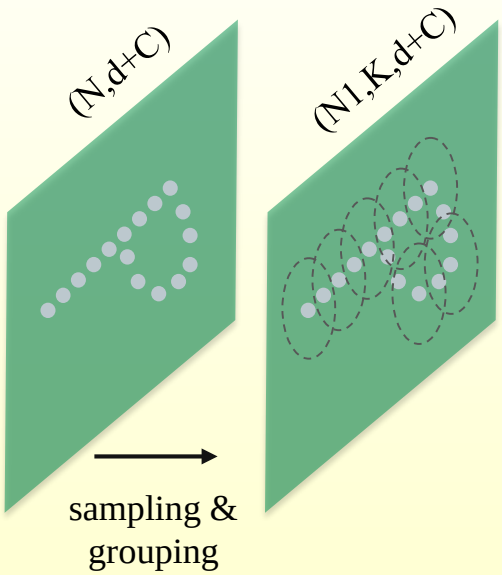
Recursively apply (shared) pointnet at local regions

✓ Hierarchical feature learning

✓ Translation invariance



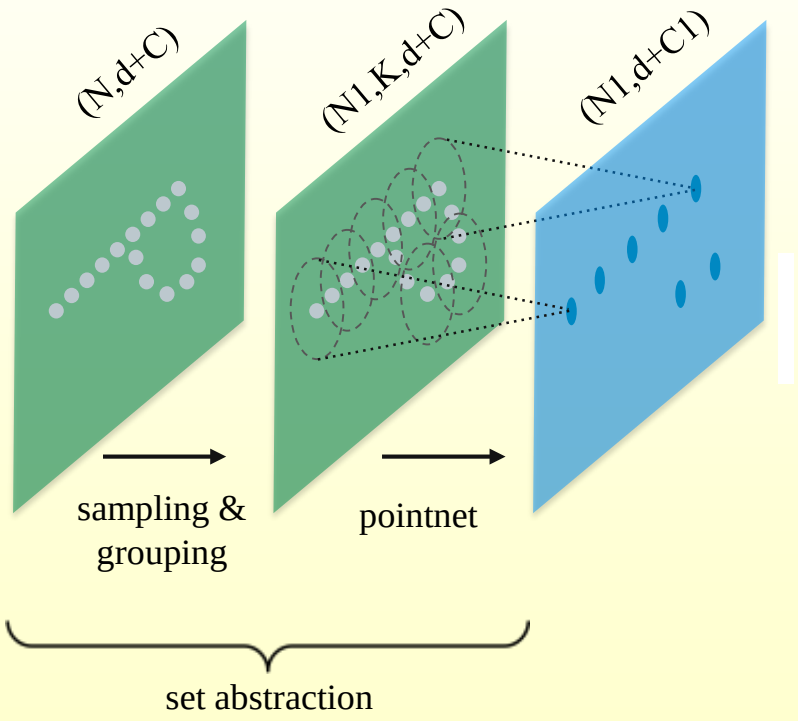
Hierarchical Point Set Feature Learning



Sampling: Farthest Point Sampling (FPS)

Grouping: radius based ball query

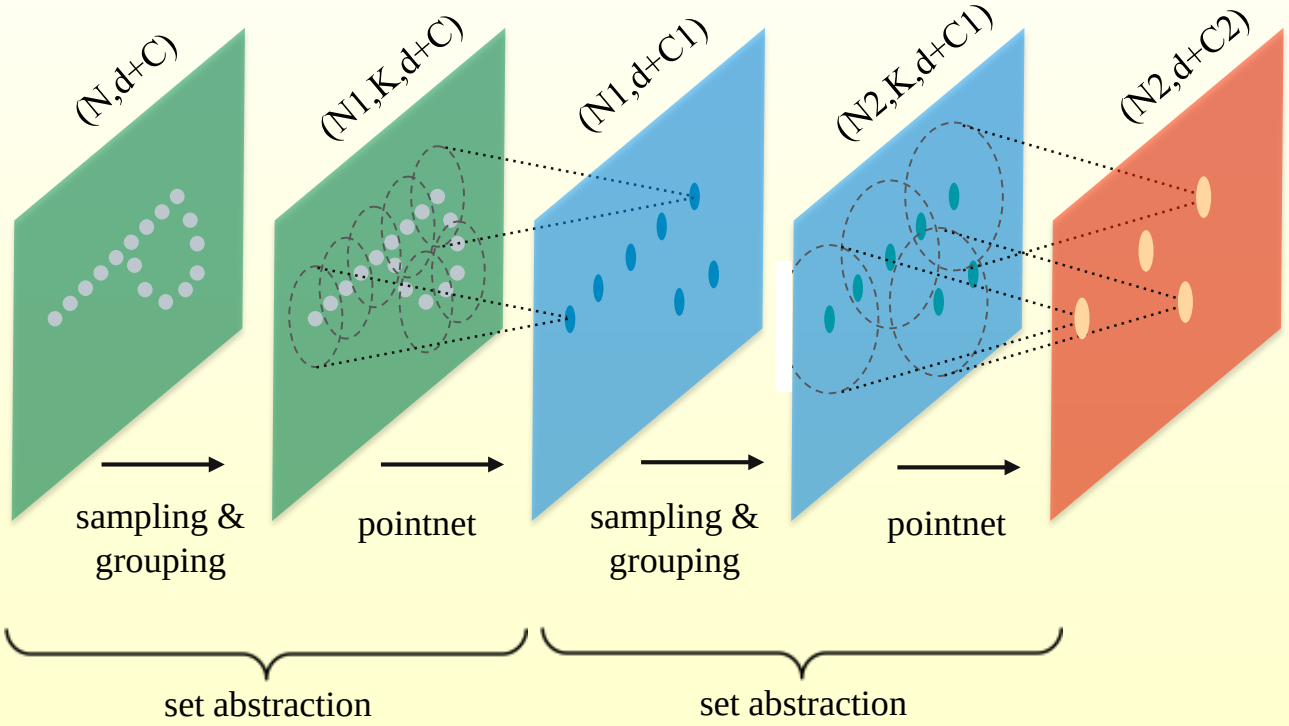
Hierarchical Point Set Feature Learning



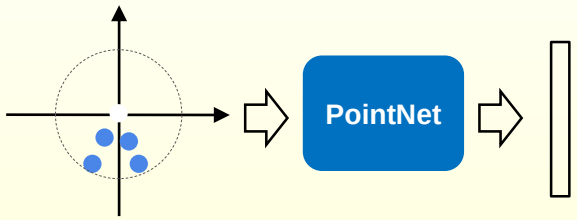
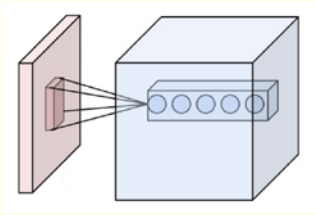
Shared pointnet applied in each local region using local coord.

Hierarchical Point Set Feature Learning

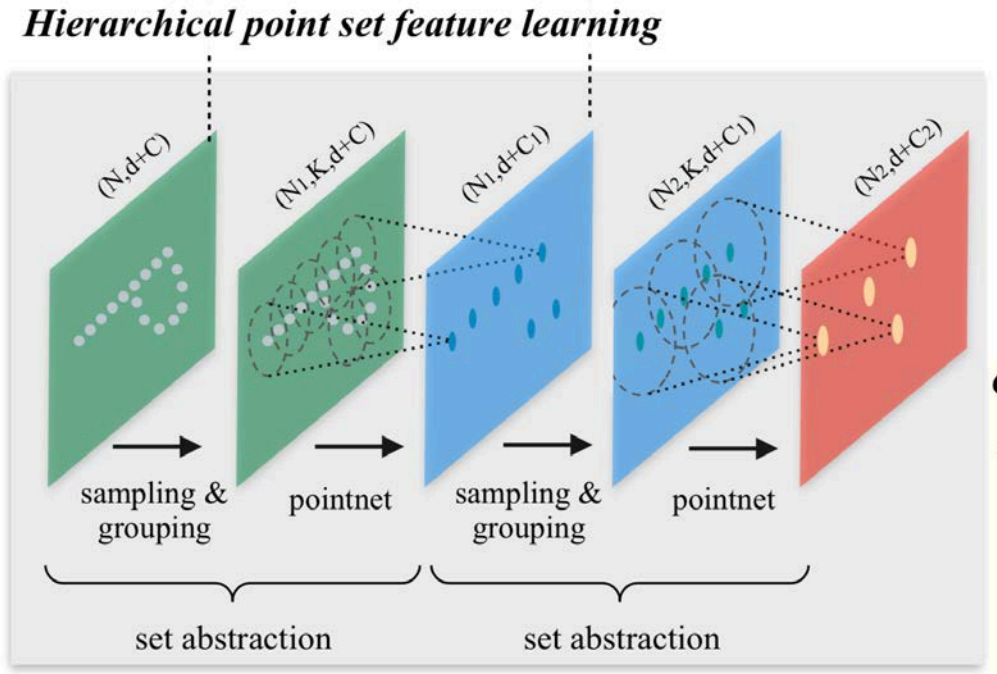
Recursively apply pointnet:

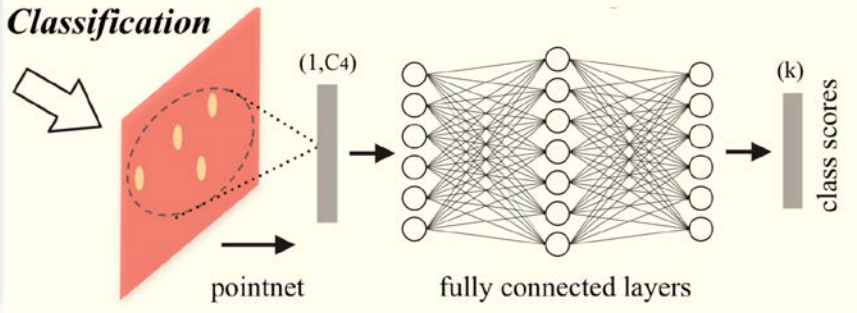


PointNet layer v.s. Convolution layer

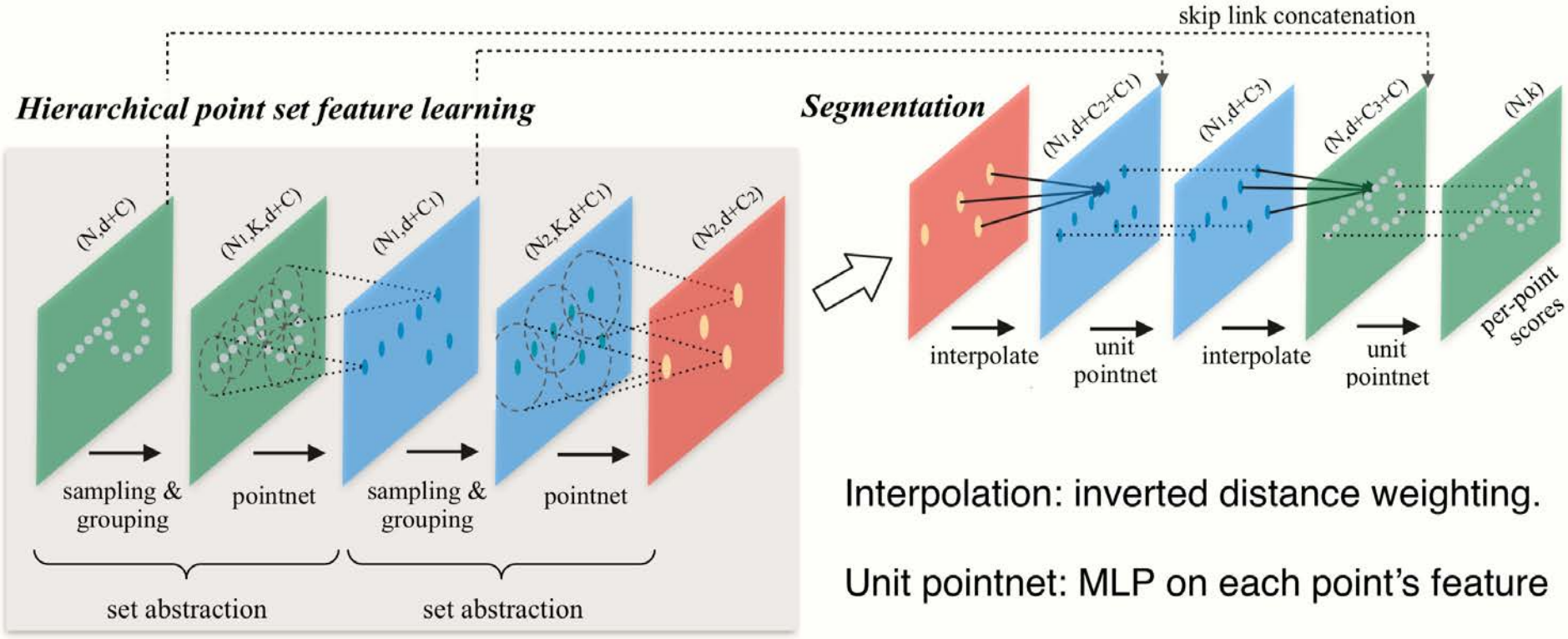
	PointNet layer	Convolution layer
		
Input:	Point set	Dense array
Operation:	PointNet (order invariant)	Convolution (index-ordered)
Neighborhood:	Radius ball query (varying #points)	Array index (fixed #pixel or #voxel)

PointNet++ for Classification and Segmentation





PointNet++ for Classification and Segmentation



Interpolation: inverted distance weighting.

Unit pointnet: MLP on each point's feature

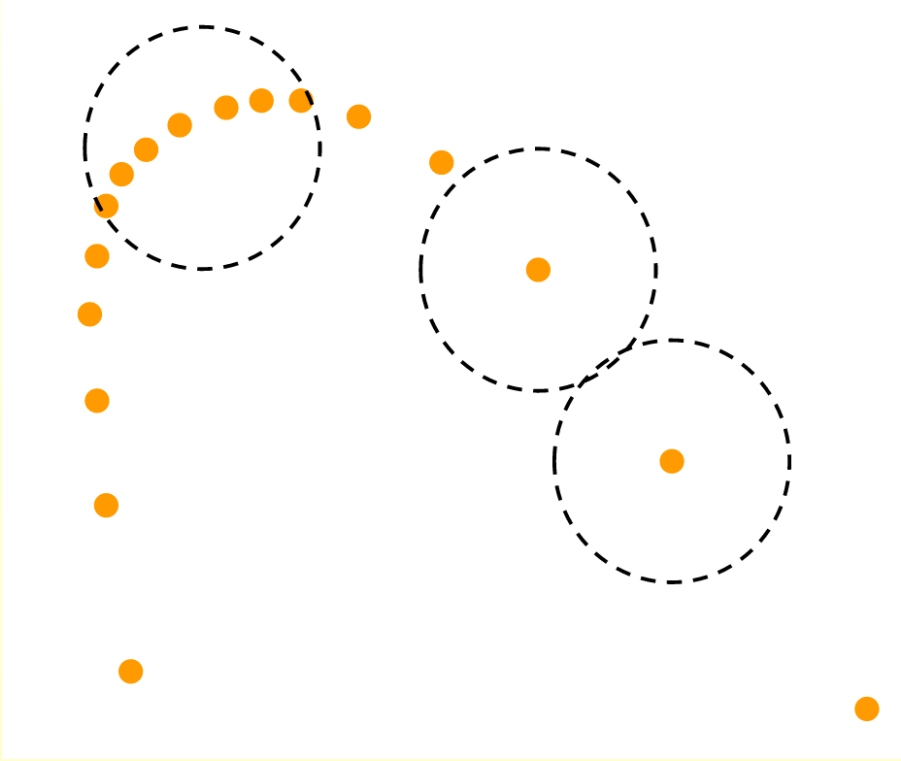
New Challenge: Non-uniform Sampling Density

Density variation is a common issue of 3D point cloud

- perspective effect, radial density variation, motion etc.

New Challenge: Non-uniform Sampling Density

The deep network need to be robust to sampling density change.



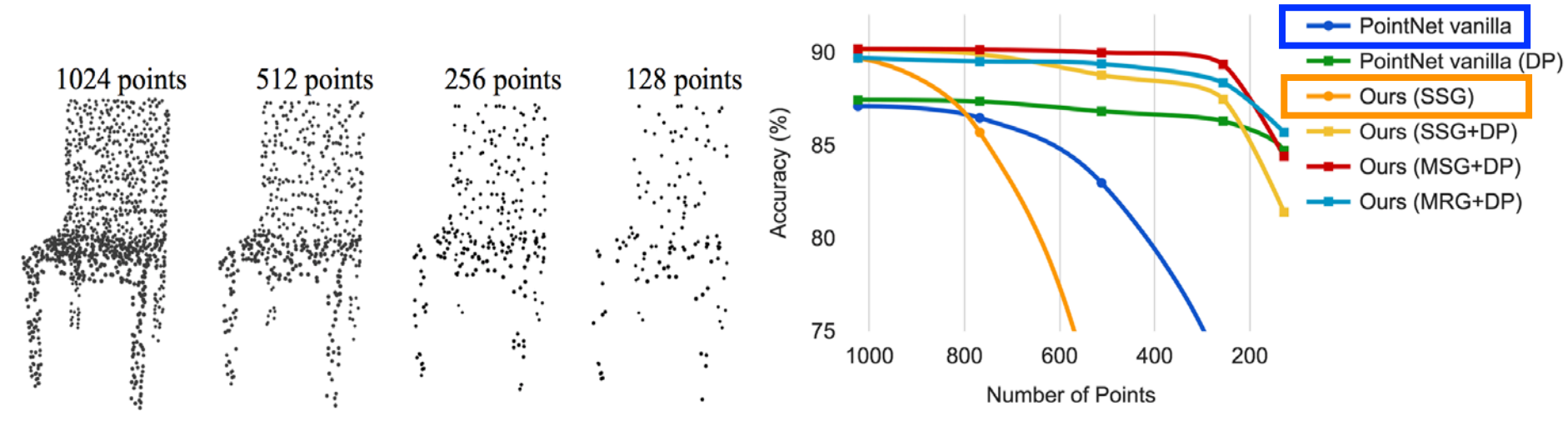
Density variation affects hierarchy

- In CNN, small kernels are usually better

Karen Simonyan & Andrew Zisserman, Very Deep Convolutional Networks for Large-scale Image Recognition, ICLR2015

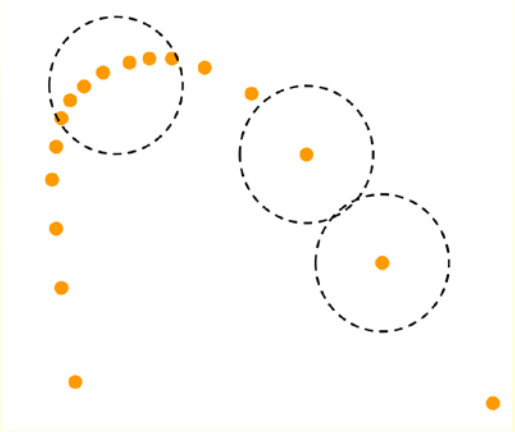
- Is it also true for point cloud learning?

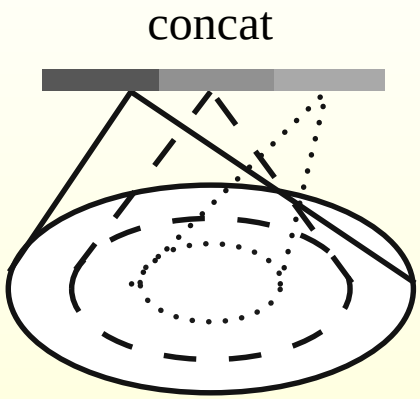
Density variation affects hierarchy

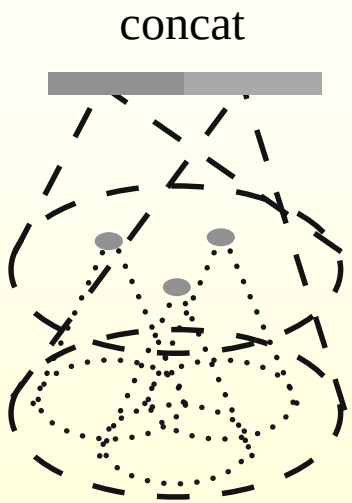


We should trust regions with high sampling density, and should use larger field of view for sparse regions.

Non-uniform Sampling Density







(a)

Multi-scale grouping (MSG)

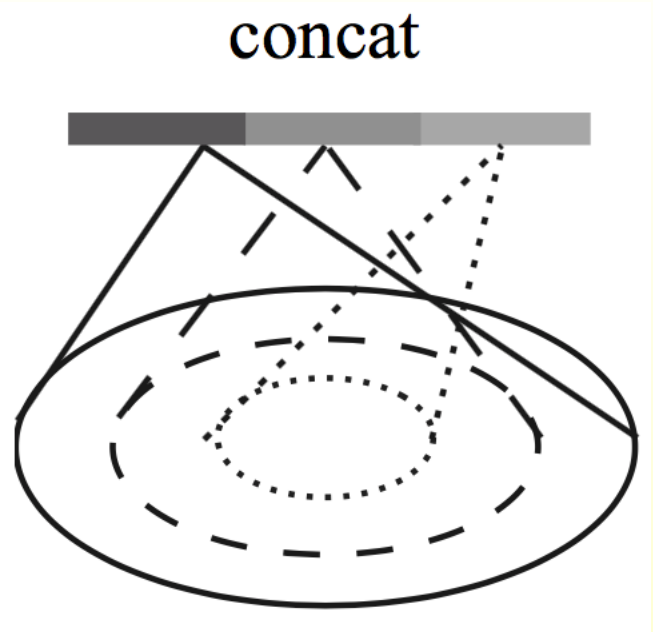
(b)

Multi-res grouping (MRG)

During Training: input point dropout with random dropout ratio

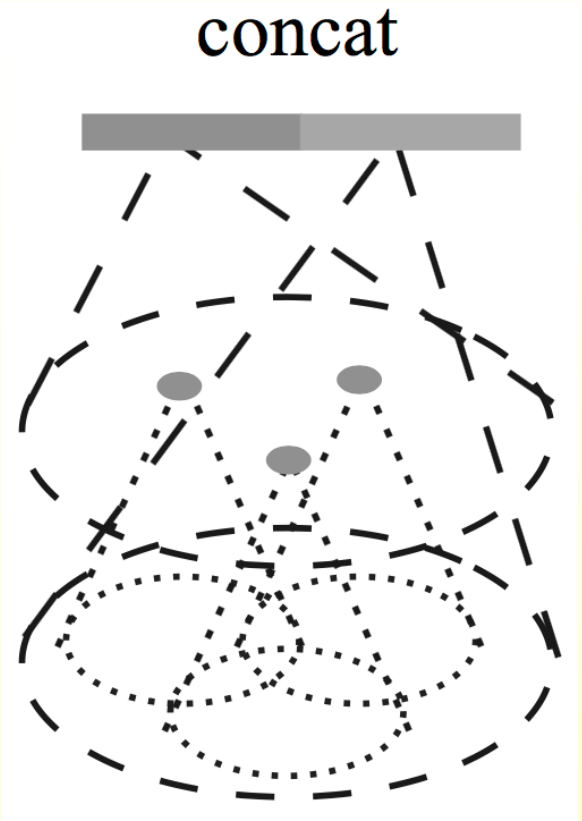
Multi-Scale Grouping (MSG)

- Extract features at multiple scales and combine them
- Add random dropout to input point cloud to simulate scanning deficiency
- Dropout ratio is sampled uniformly in [0, 1]

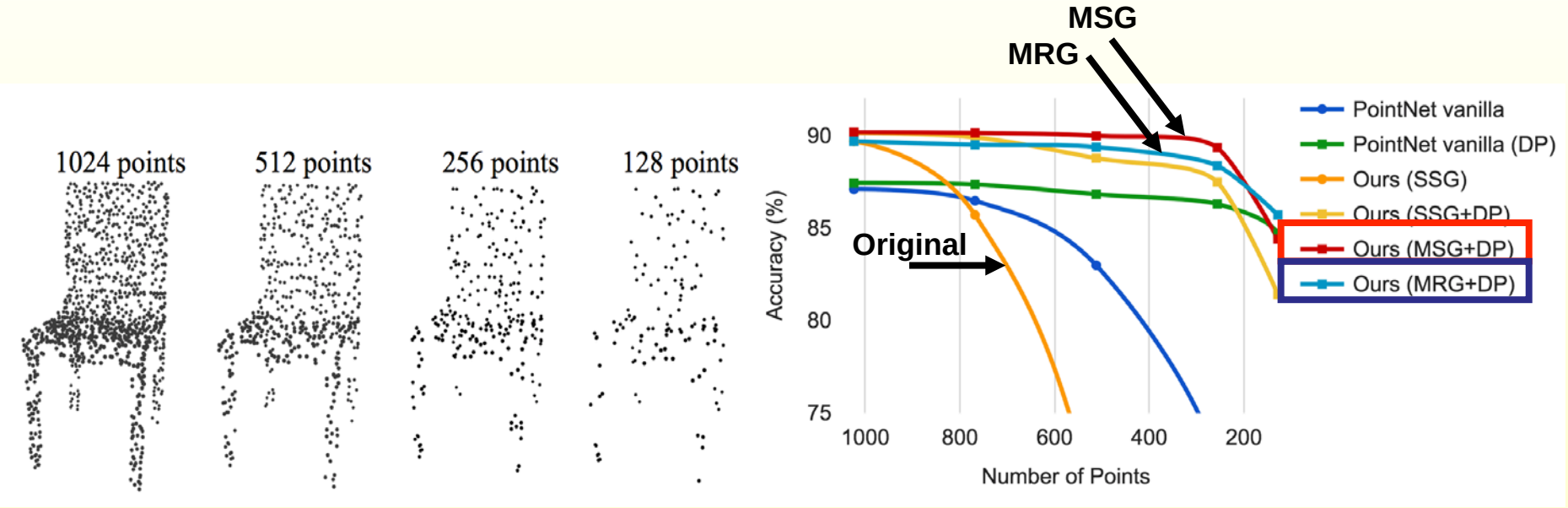


Multi-Resolution Grouping (MRG)

- Drawback of MSG: expensive
  - Need to run PointNet on many neighborhoods
- Multi-resolution grouping: reuse the computation from different levels

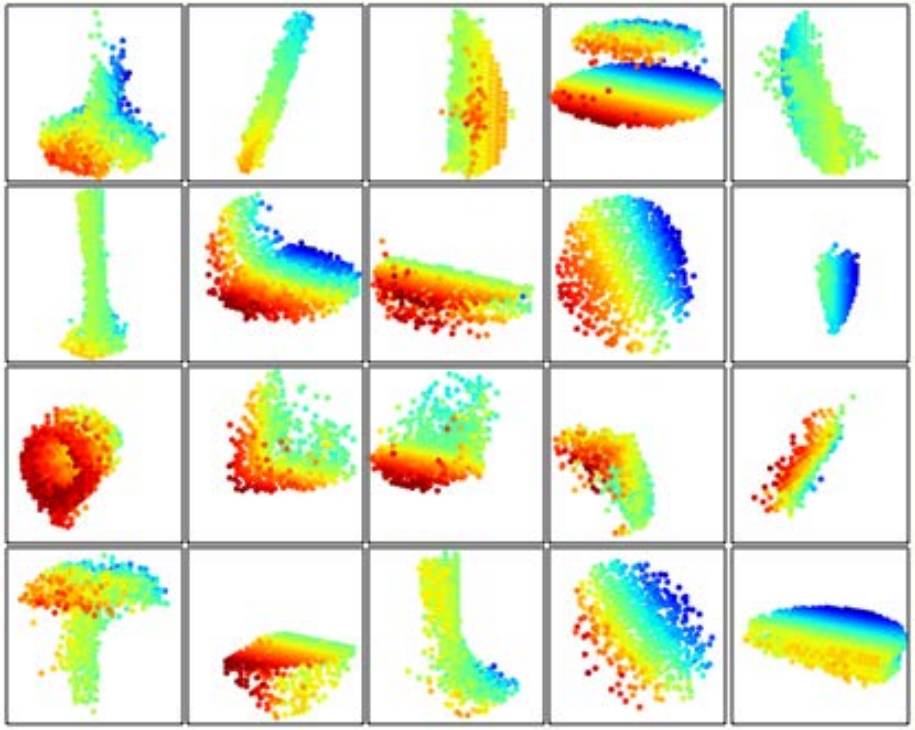


Robust learning under varying sampling density



PointNet++ Feature Visualization

Trained on ModelNet40 (mostly furniture) we see structures of planes, double planes, lines, corners etc.

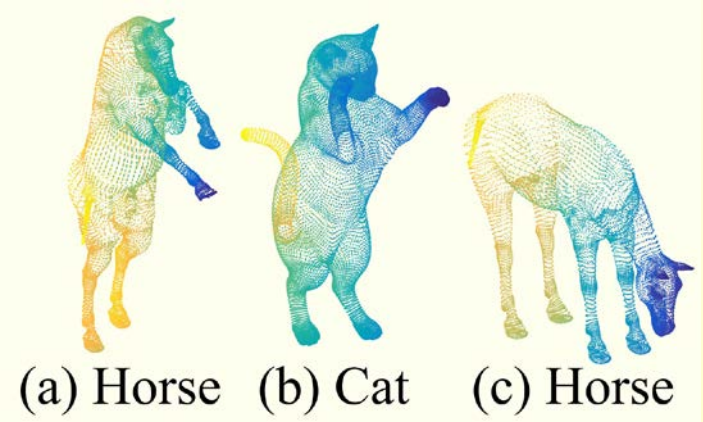


First layer patterns

PointNet++ Results: Non-Euclidean Space

For organic shape recognition, PointNet++ can generalize to non-Euclidean space:

- intrinsic point features (HKS, WKS, Gaussian curvature)
- intrinsic distance metric (geodesic)



(a) Horse (b) Cat (c) Horse

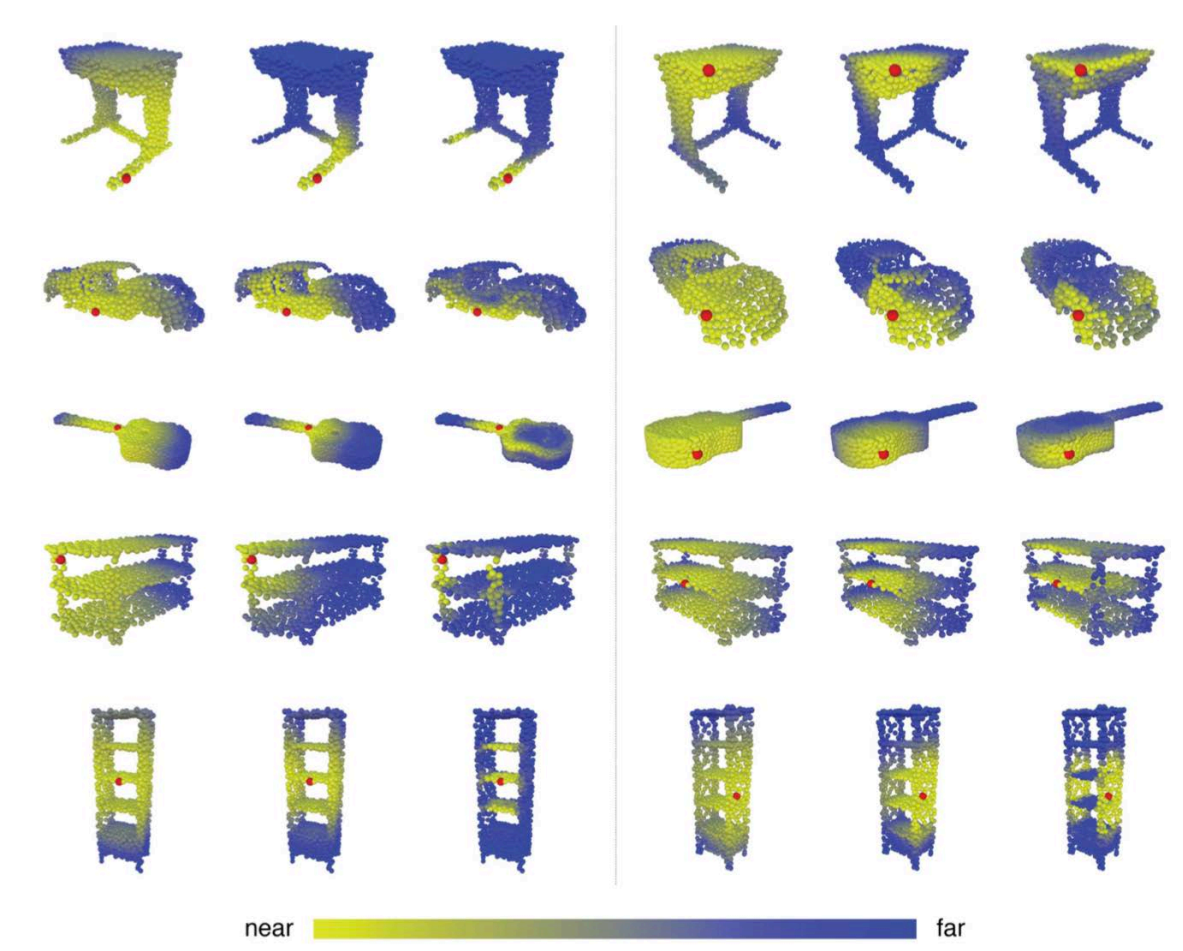
	Metric space	Input feature	Accuracy (%)
DeepGM [13]	-	Intrinsic features	93.03
Ours	Euclidean	XYZ	60.18
	Euclidean	Intrinsic features	94.49
	Non-Euclidean	Intrinsic features	**96.09**

Table 3: SHREC15 Non-rigid shape classification.

Dynamic Graph CNN [Wang et al. 2018]

- **KNN graph. Notion of "nearby" changes each layer**

Point-to-point distance per layer:



Summary

PointNet, a novel type of **neural network that directly consumes point clouds** and well respects permutation invariance.

In PointNet++, pointnet is used as a base learning module, to achieve **hierarchical feature learning** as well as **robustness to varying sampling density**.

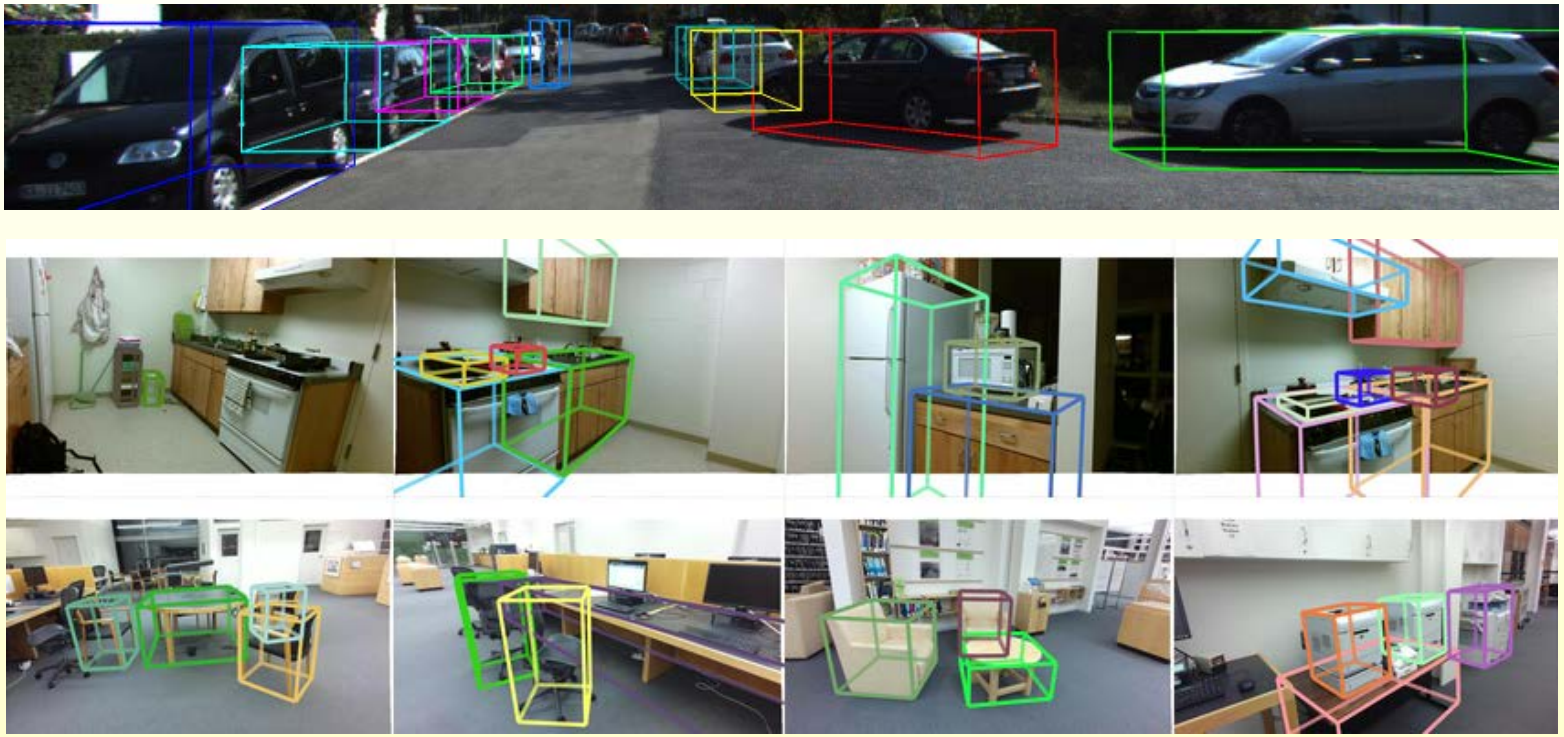
In Dynamic Graph CNN, on can use feature space for neighborhood search.

Agenda

- Deep Nets for Point Cloud Analysis
  - PointNet, PointNet++, Dynamic Graph CNN
  - **3D Object Detection with PointNets**
- Deep Nets for Point Cloud Generation

PointNets for 3D Object Detection

What is 3D Object Detection?



What is 3D Object Detection?

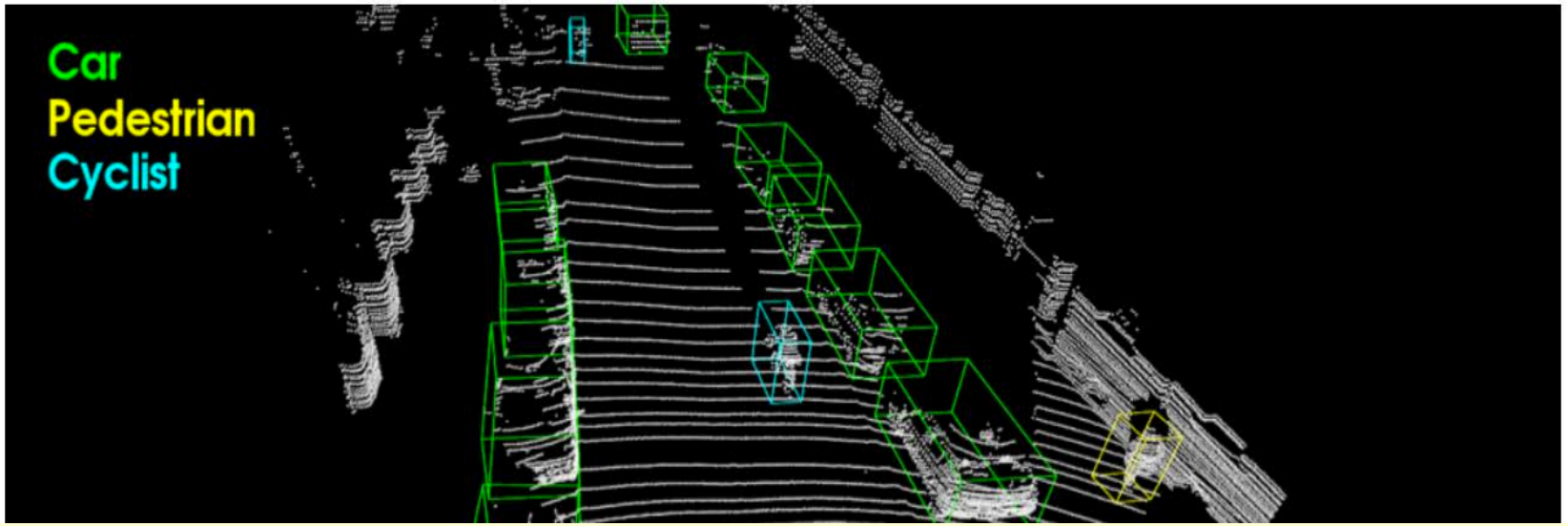
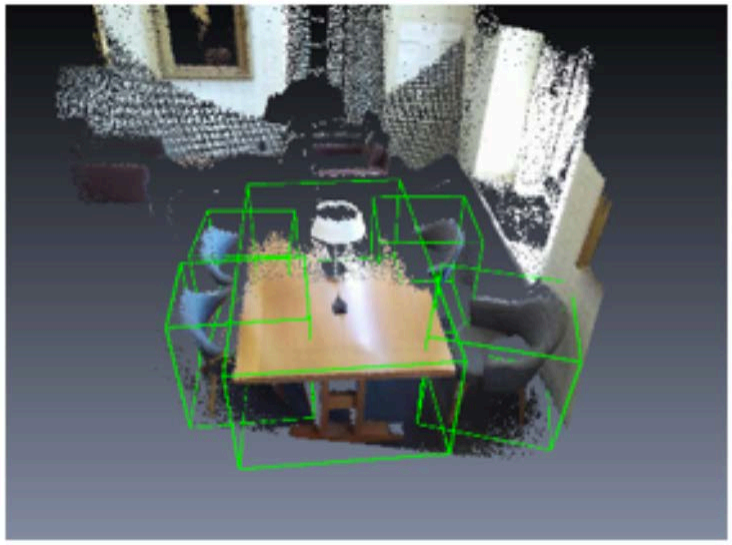


Figure from the recent VoxelNet paper from Apple.

What is 3D Object Detection?



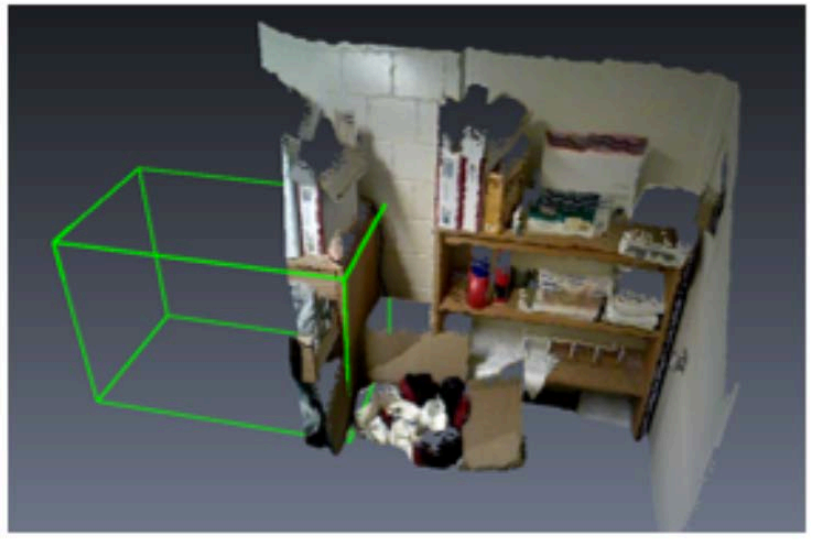
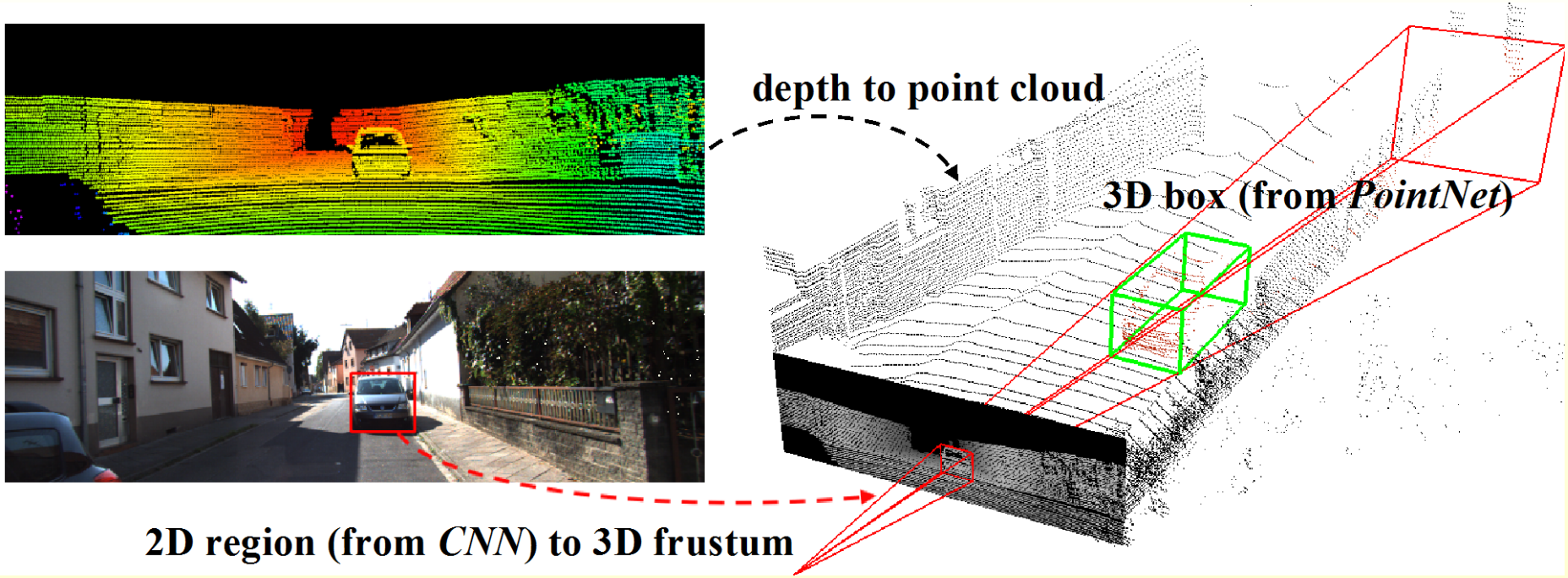


Figure from ICCV17 paper 2d-driven 3d object detection.

Key Problem in 3D Object Detection

- How to localize 3D objects? – large search space
- How to represent 3D data? – projection to image, voxelization or raw point clouds

Frustum PointNets for 3D Object Detection



+ ***Leveraging mature 2D detectors*** *for region proposal. 3D search space reduced.*

+ Solving 3D detection problem with ***3D data and 3D deep learning*** *architectures.*

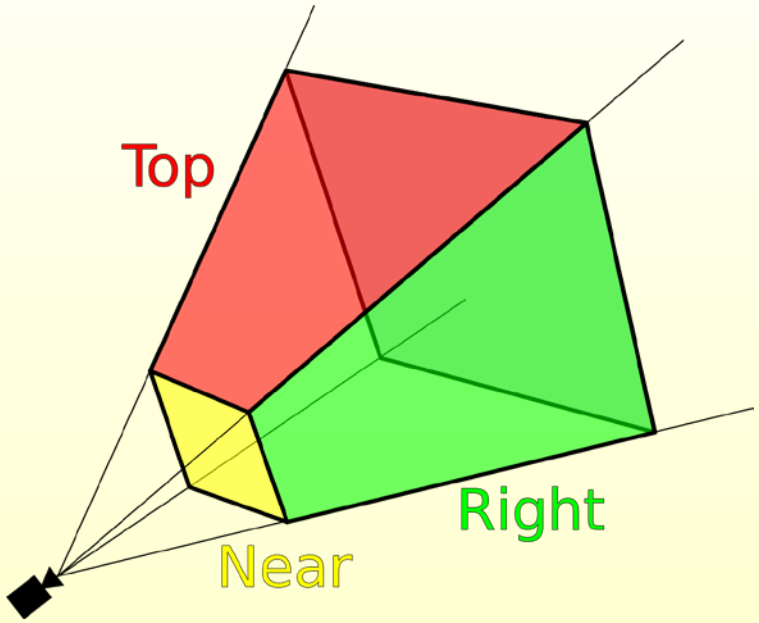
Frustum-based 3D Object Detection

Benefits:

- Leverage 2D detector to greatly reduce the 3D search space.
- Direct 3D estimation in 3D point cloud. No more projections.

Challenges:

- Occlusions and clutters are common in frustum point cloud.
- Largely varying ranges of points in frustums.



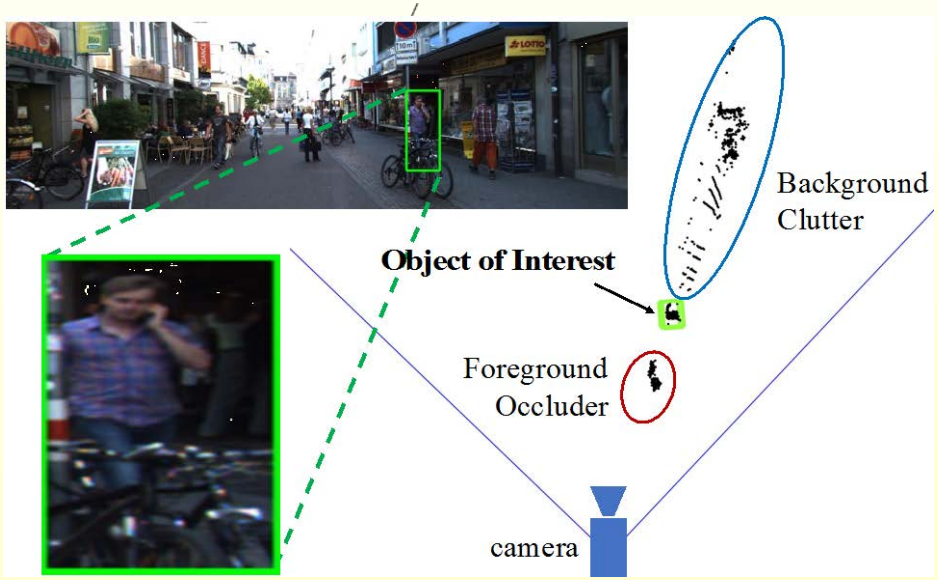
Frustum-based 3D Object Detection

Benefits:

- Leverage 2D detector to greatly reduce the 3D search space.
- Direct 3D estimation in 3D point cloud. No more projections.

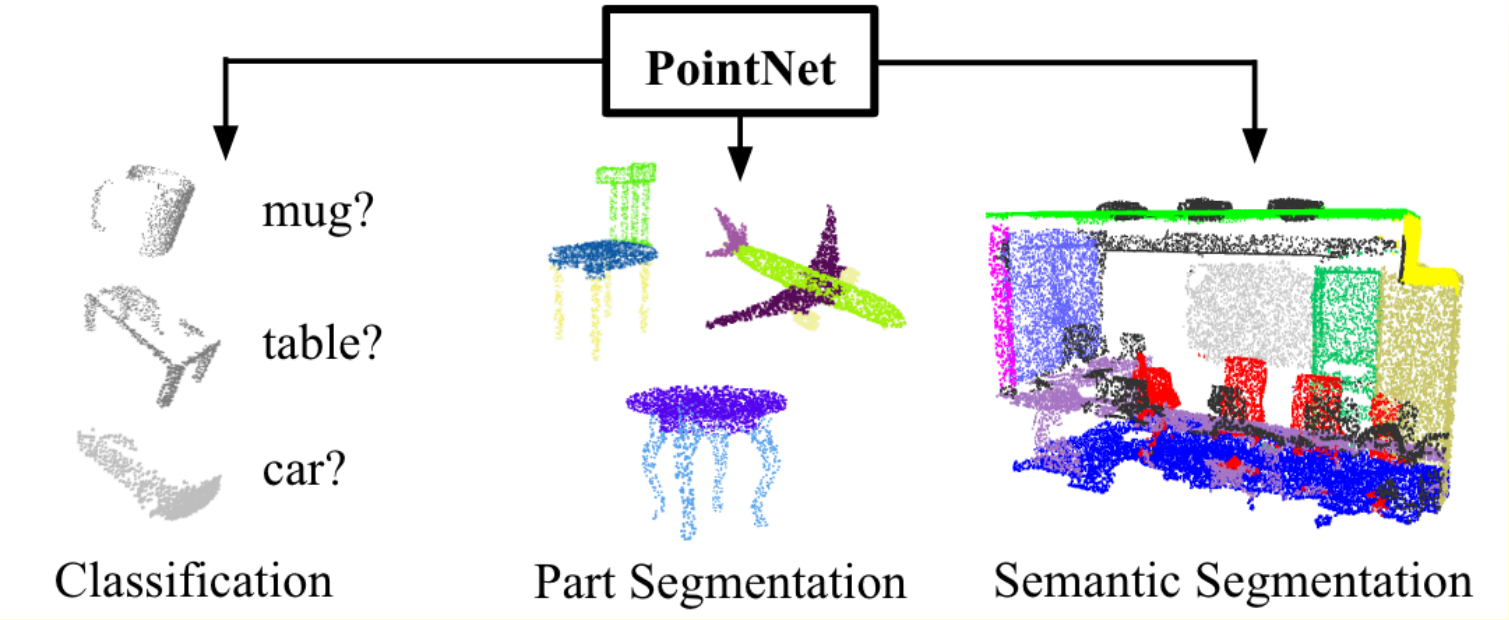
Challenges:

- Occlusions and clutters are common in frustum point cloud.
- Largely varying ranges of points in frustums.



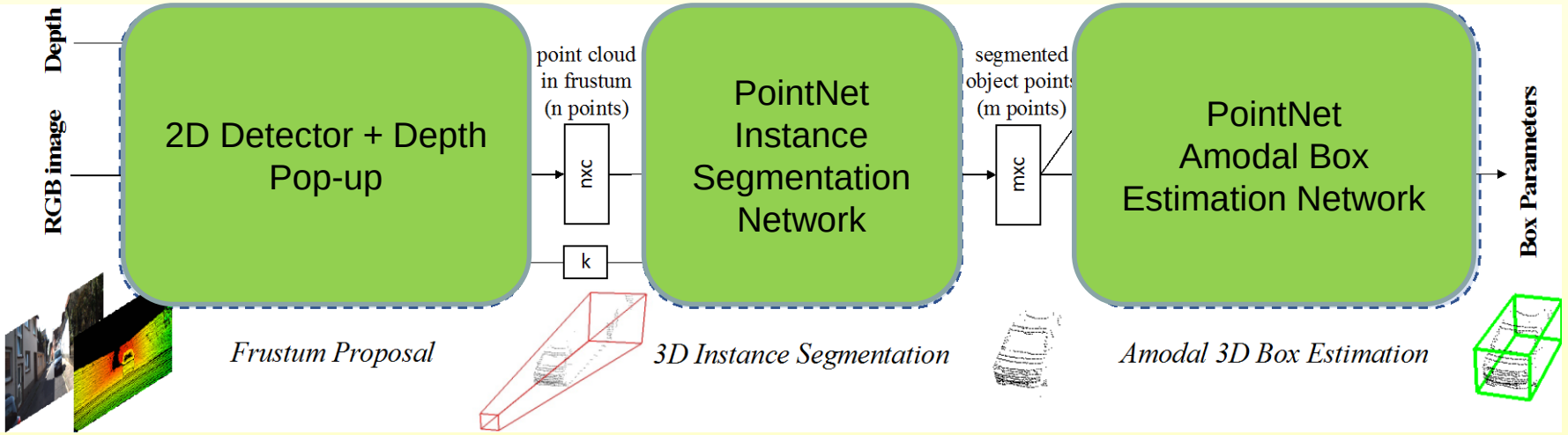
Frustum PointNets

- Use PointNet architectures for data-driven object detection in Frustums.

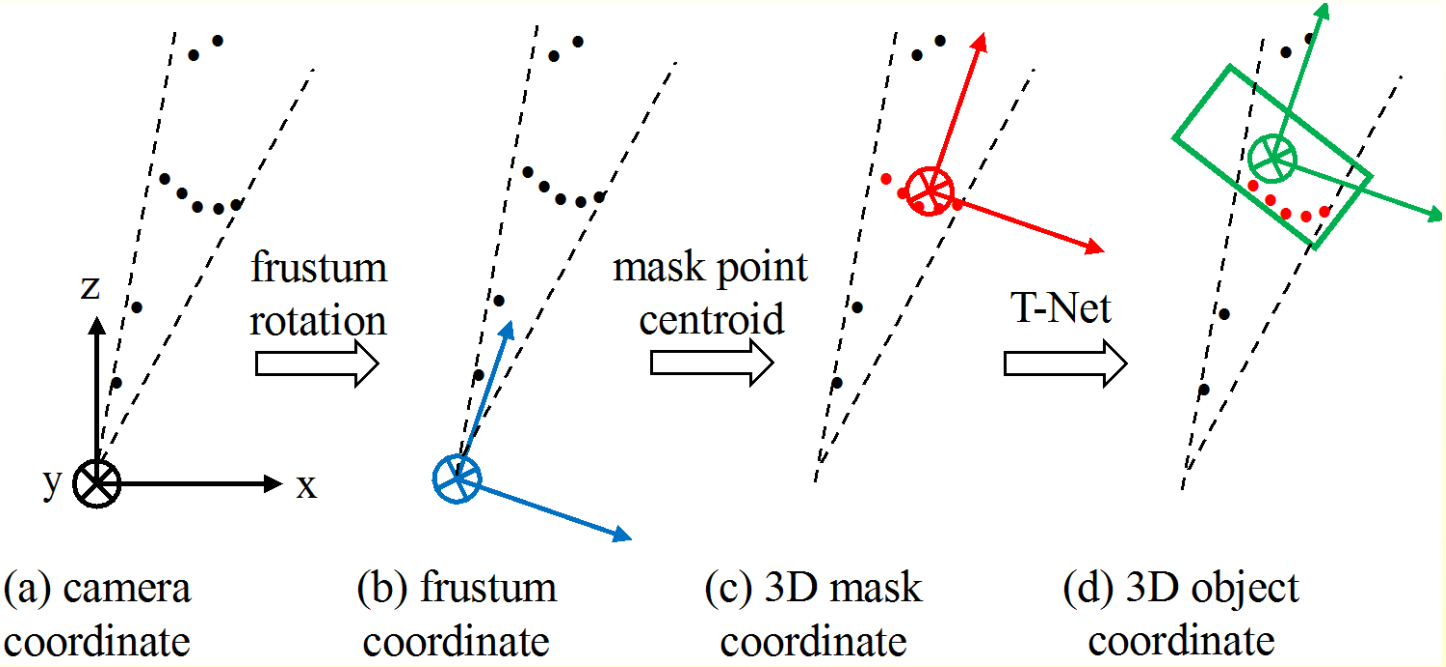


Frustum PointNets

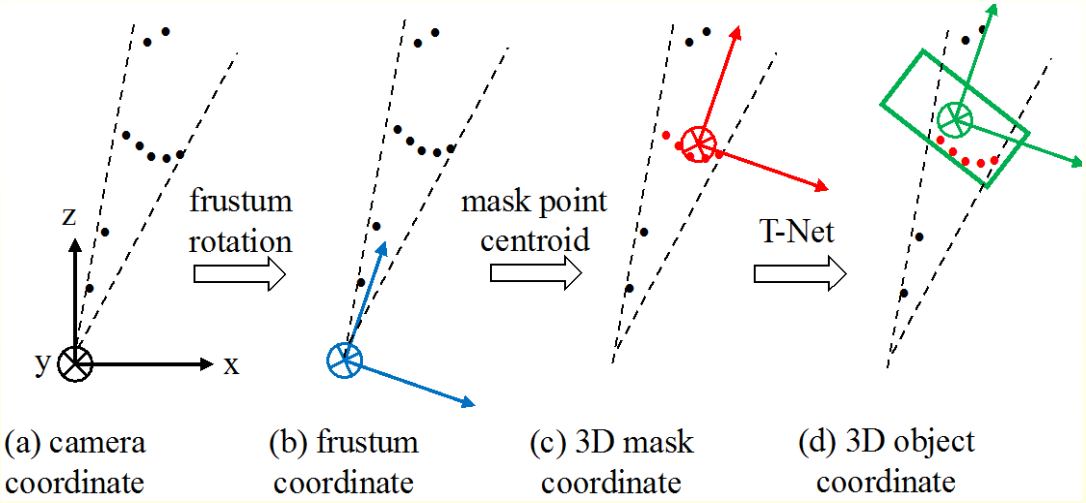
- Use PointNet architectures for data-driven object detection in Frustums.
- The complete 3D detection pipeline is decomposed into three key modules.



Coordinates Normalization



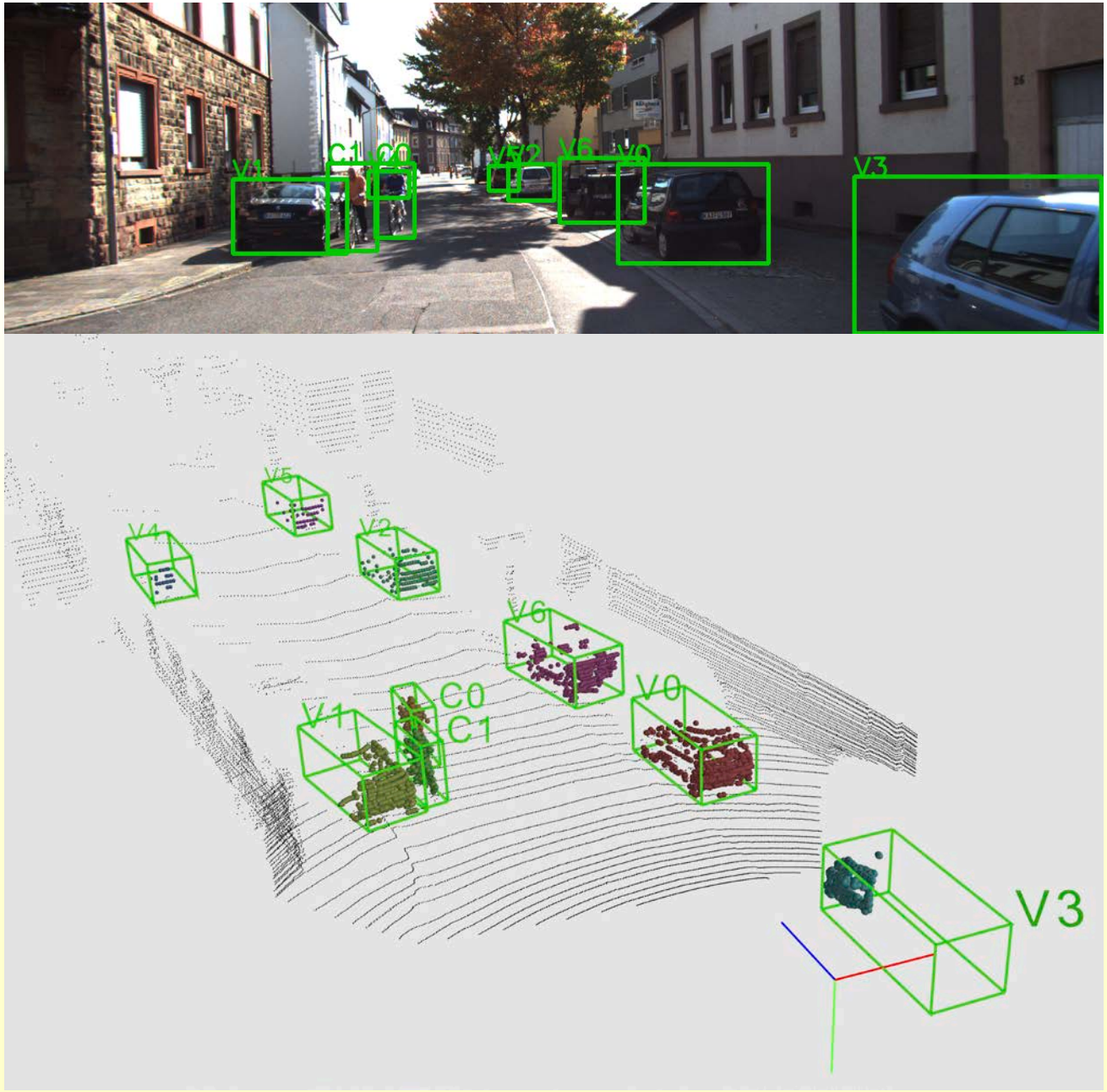
Coordinate Normalization



frustum rot.	mask centralize	t-net	accuracy
-	-	-	12.5
√	-	-	48.1
-	√	-	64.6
√	√	-	71.5
√	√	√	**74.3**

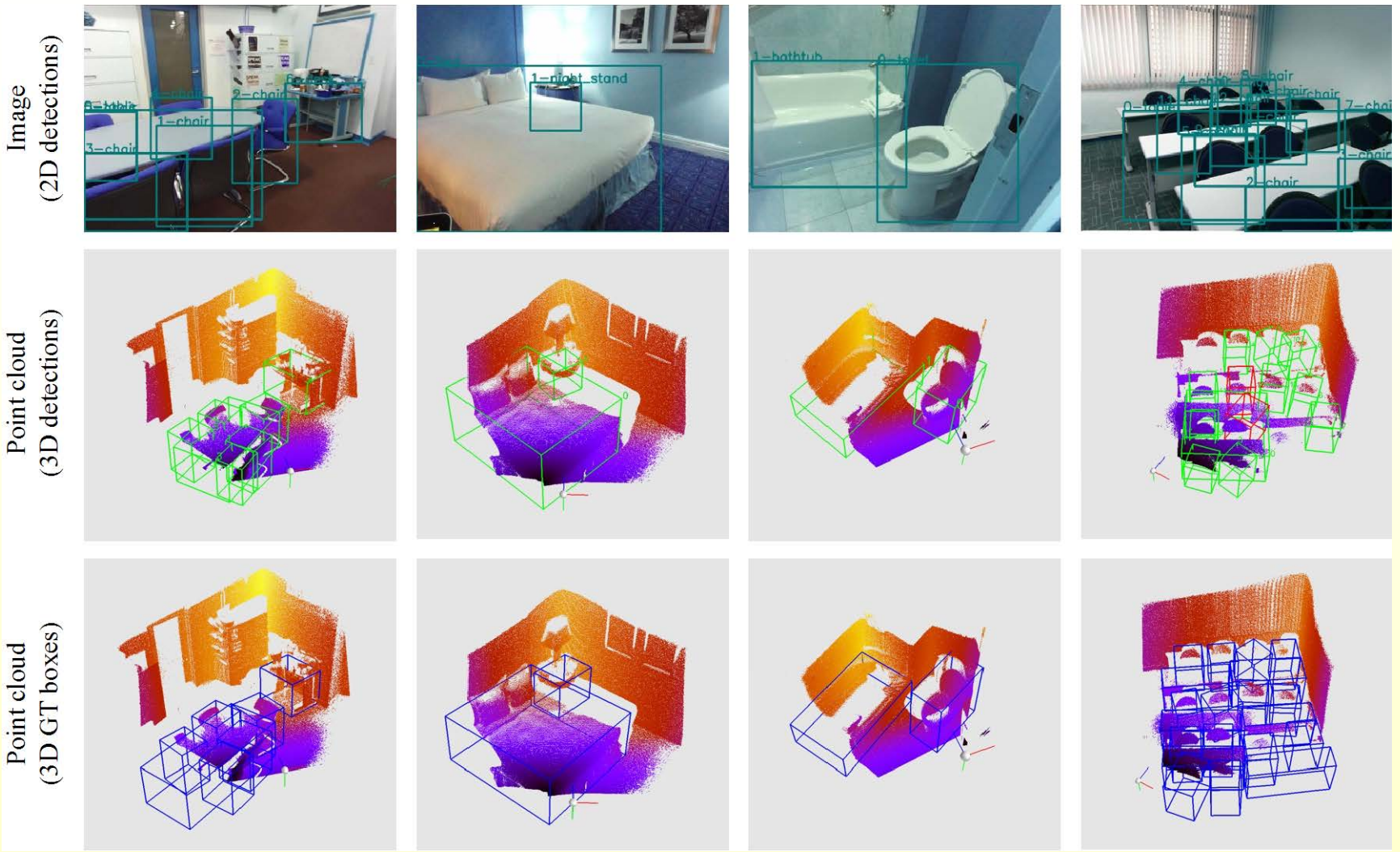
Table 7. **Effects of point cloud normalization.** Metric is 3D box estimation accuracy with IoU=0.7.

KITTI Results:



Remarkable box estimation accuracy even with a dozen of points or with very partial point cloud

SUN-RGBD Results:



Agenda

- Deep Nets for Point Cloud Analysis
  - PointNet, PointNet++, Dynamic Graph CNN
  - 3D Object Detection with PointNets
- **Deep Nets for Point Cloud Generation**

Point Set Generation Network

How to generate point clouds with a deep neural network?

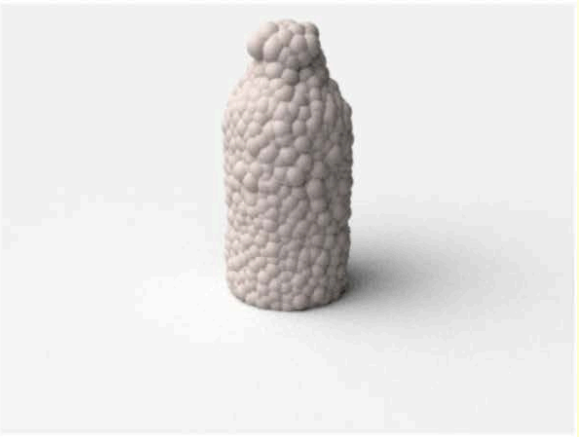
- **3D reconstruction from a single image**
- Point cloud autoencoder
- Point cloud GAN (generative adversarial network)

3D Perception from a Single Image

3D Reconstruction from Real Images



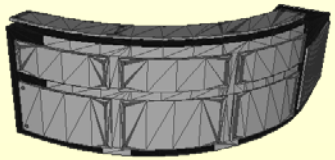


Generate synthetic data for learning

Image

rendering



3D model

sampling

$$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \dots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$$

Groundtruth point cloud

Pipeline

Image

Deep Neural Network

Predicted set

$$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \dots \\ (x_n, y_n, z_n) \end{array} \right\}$$

$$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \dots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$$

Groundtruth point cloud

Pipeline

Image

Deep Neural Network

Predicted set

$$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \dots \\ (x_n, y_n, z_n) \end{array} \right\}$$

Groundtruth point cloud

$$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \dots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$$

Point Set

Distance

Pipeline

Image

Deep Neural Network

Predicted set

$$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \dots \\ (x_n, y_n, z_n) \end{array} \right\}$$

Groundtruth point cloud

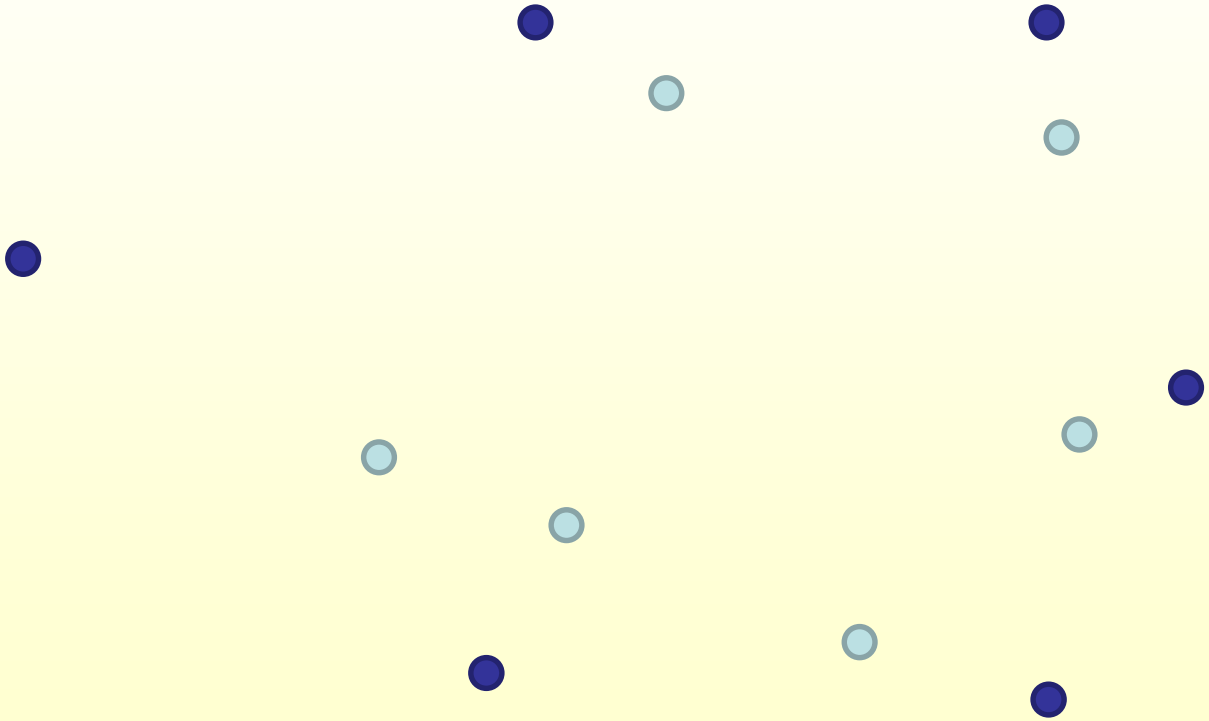
$$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \dots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$$

Point Set

Distance

Distance metrics between point sets

Given two sets of points, measure their discrepancy

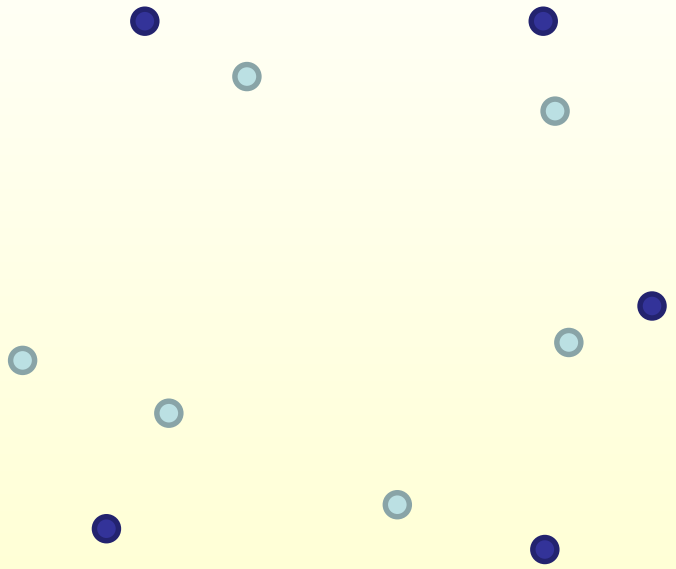


Common distance metrics

Worst case: Hausdorff distance (HD)

Average case: Chamfer distance (CD)

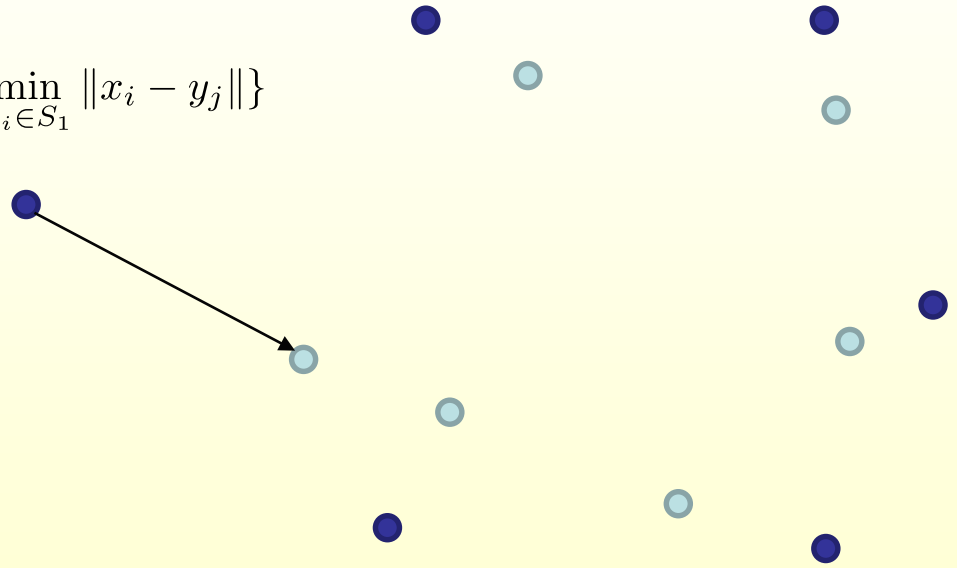
Optimal case: Earth Mover's distance (EMD)



Common distance metrics

Worst case: Hausdorff distance (HD)

$$d_{\mathrm{HD}}(S_1, S_2) = \max\{\max_{x_i \in S_1} \min_{y_j \in S_2} \|x_i - y_j\|, \max_{y_j \in S_2} \min_{x_i \in S_1} \|x_i - y_j\|\}$$



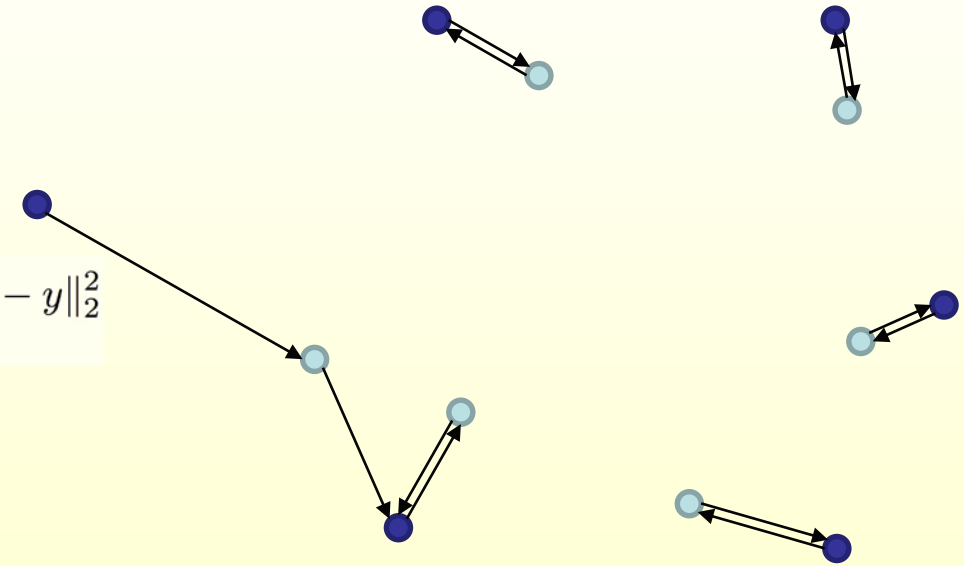
A single farthest pair determines the distance.
In other words, ***not robust to outliers!***

Common distance metrics

Worst case: Hausdorff distance (HD)

Average case: Chamfer distance (CD)

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



Average all the nearest neighbor distance by nearest neighbors

Common distance metrics

Worst case: Hausdorff distance (HD)

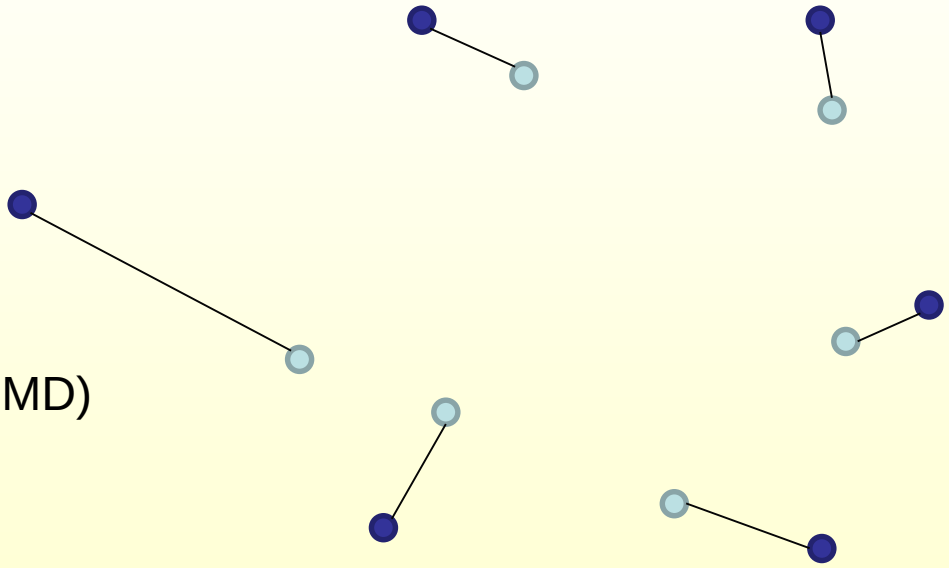
Average case: Chamfer distance (CD)

Optimal case: Earth Mover's distance (EMD)

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

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

Solves the optimal transportation (bipartite matching) problem!



Pipeline

Image

Deep Neural Network

Predicted set

$$\left\{ \begin{array}{c} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \dots \\ (x_n, y_n, z_n) \end{array} \right\}$$

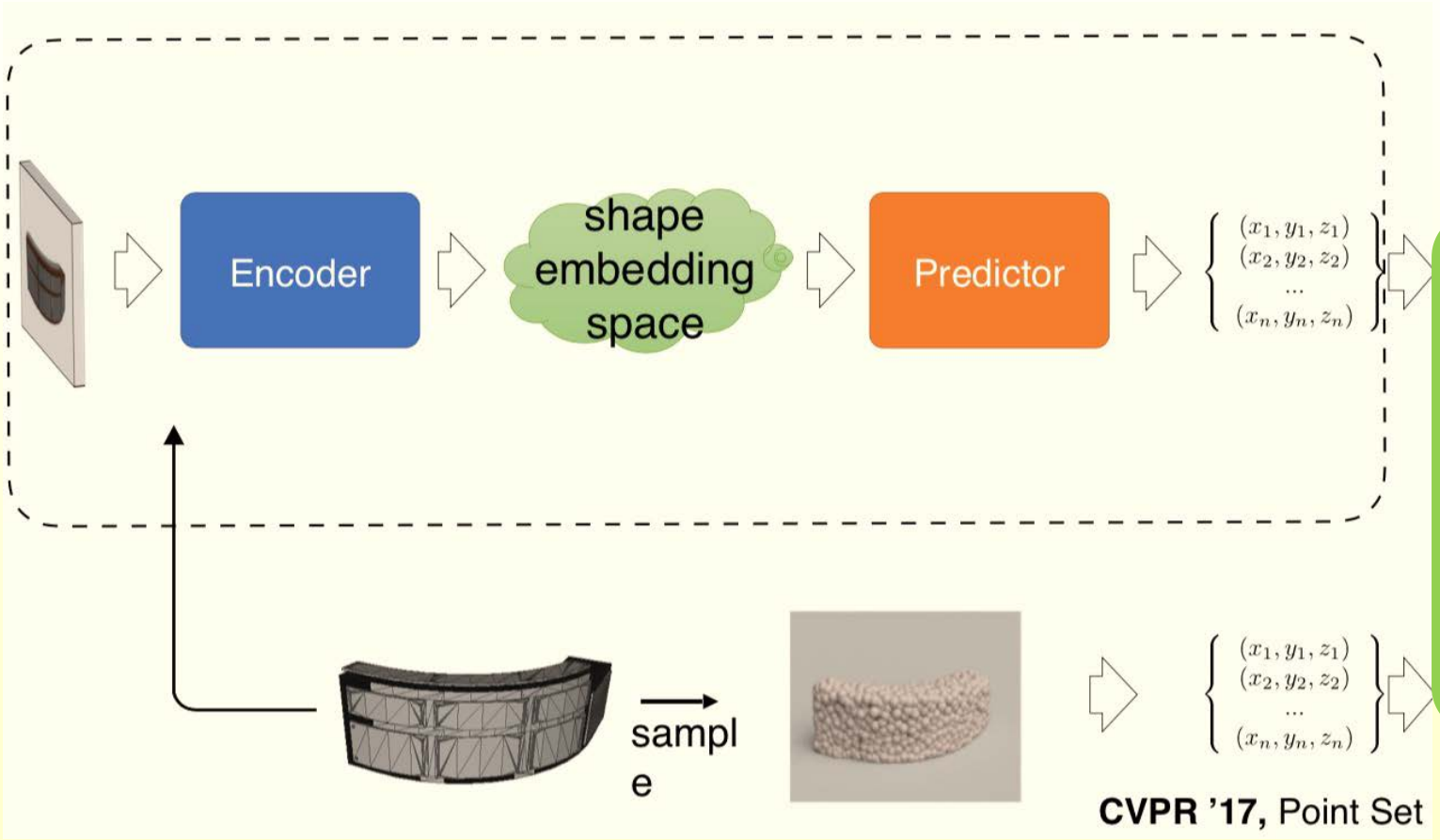
Groundtruth point cloud

$$\left\{ \begin{array}{c} (x'_1, y'_1, z'_1) \\ (x'_2, y'_2, z'_2) \\ \dots \\ (x'_n, y'_n, z'_n) \end{array} \right\}$$

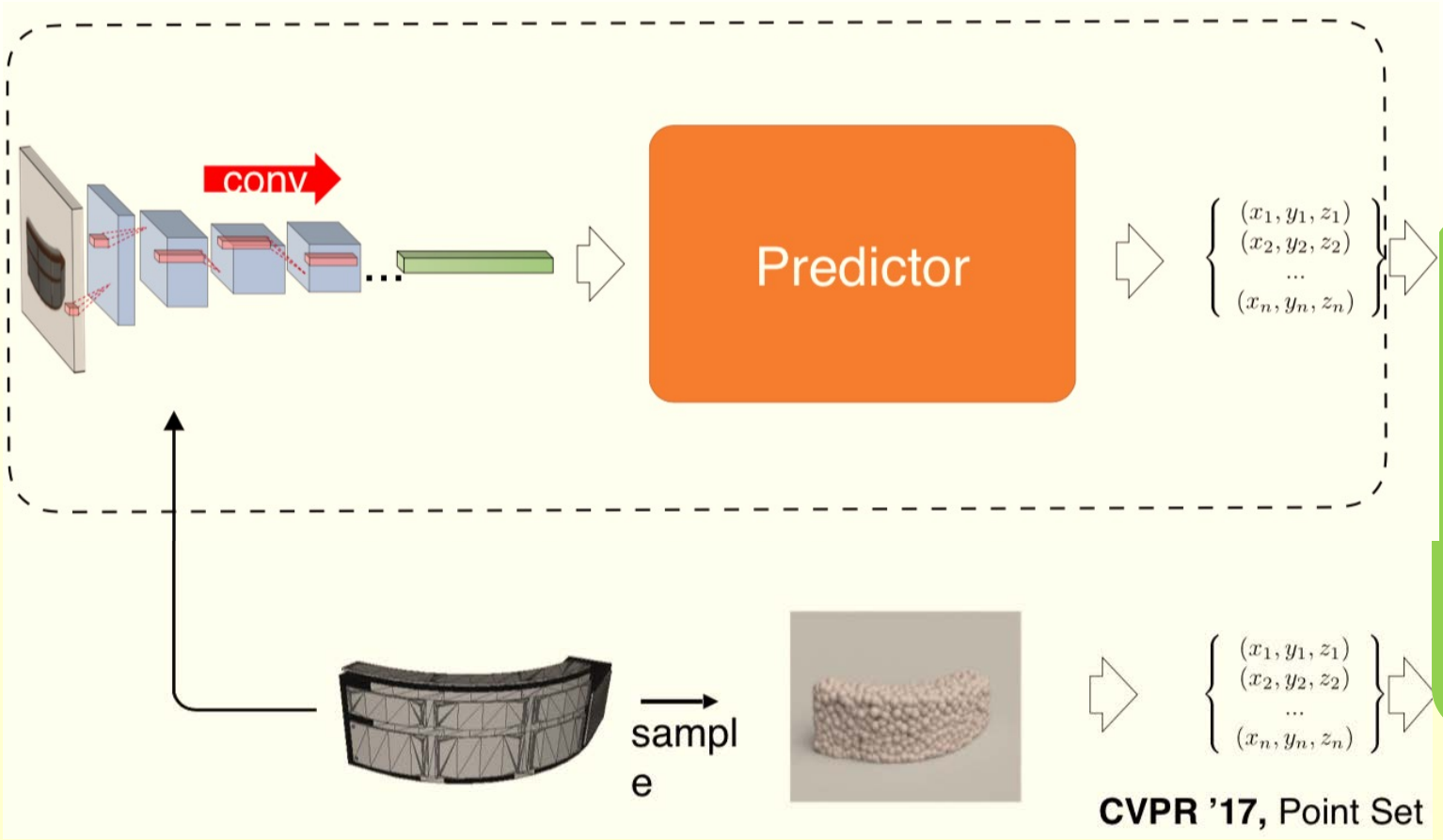
Point Set

Distance

Pipeline


Encoder
shape embedding space
Predictor
(x_1, y_1, z_1)
(x_2, y_2, z_2)
...
(x_n, y_n, z_n)
Point Set
Distance
sampl e
(x_1, y_1, z_1)
(x_2, y_2, z_2)
...
(x_n, y_n, z_n)
CVPR '17, Point Set

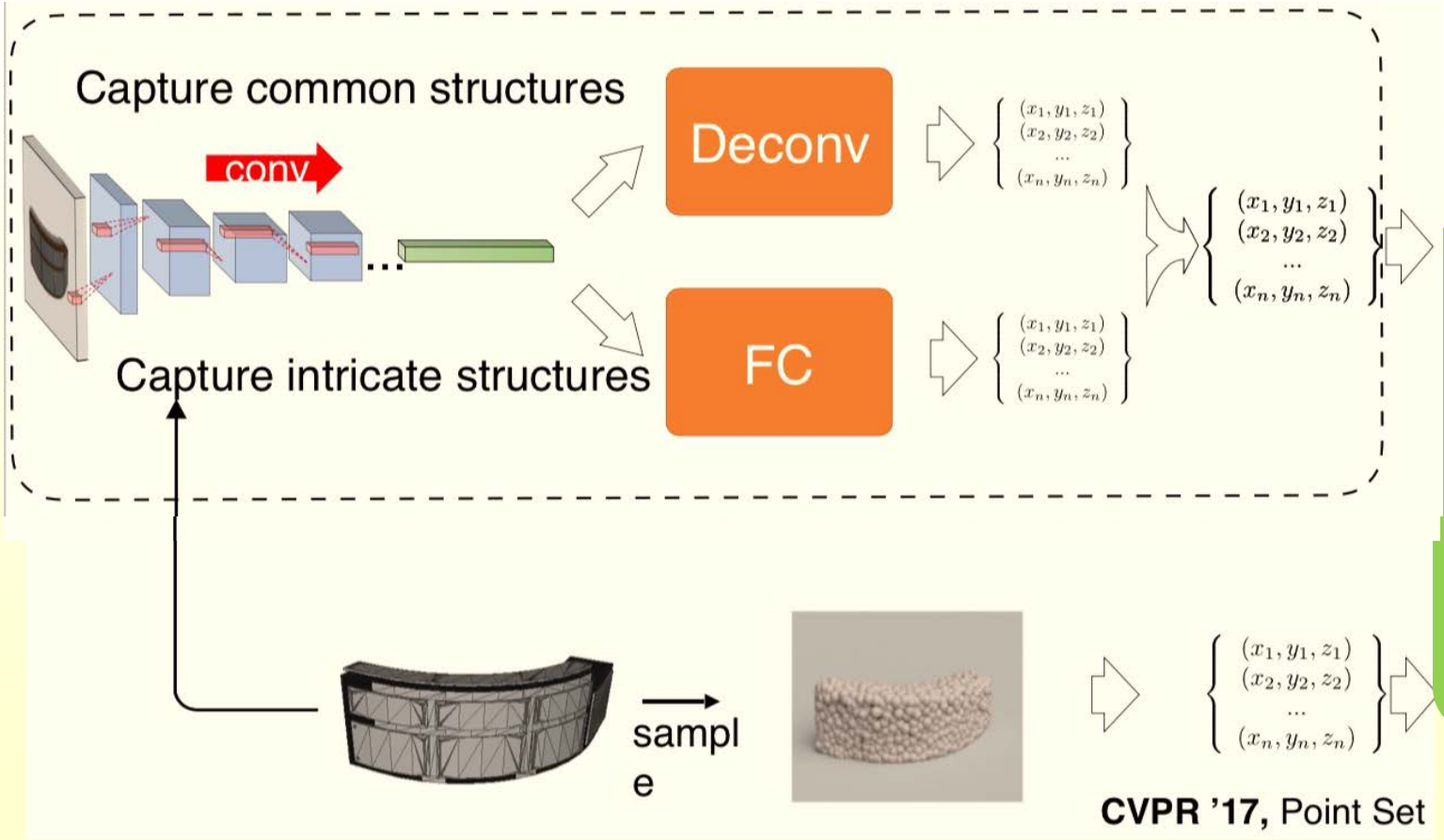
Pipeline


conv
Predictor
(x_1, y_1, z_1)
(x_2, y_2, z_2)
...
(x_n, y_n, z_n)
Point Set
Distance
sampl
e
(x_1, y_1, z_1)
(x_2, y_2, z_2)
...
(x_n, y_n, z_n)
CVPR '17, Point Set

Natural statistics of geometry

- Many local structures are common
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates

Pipeline



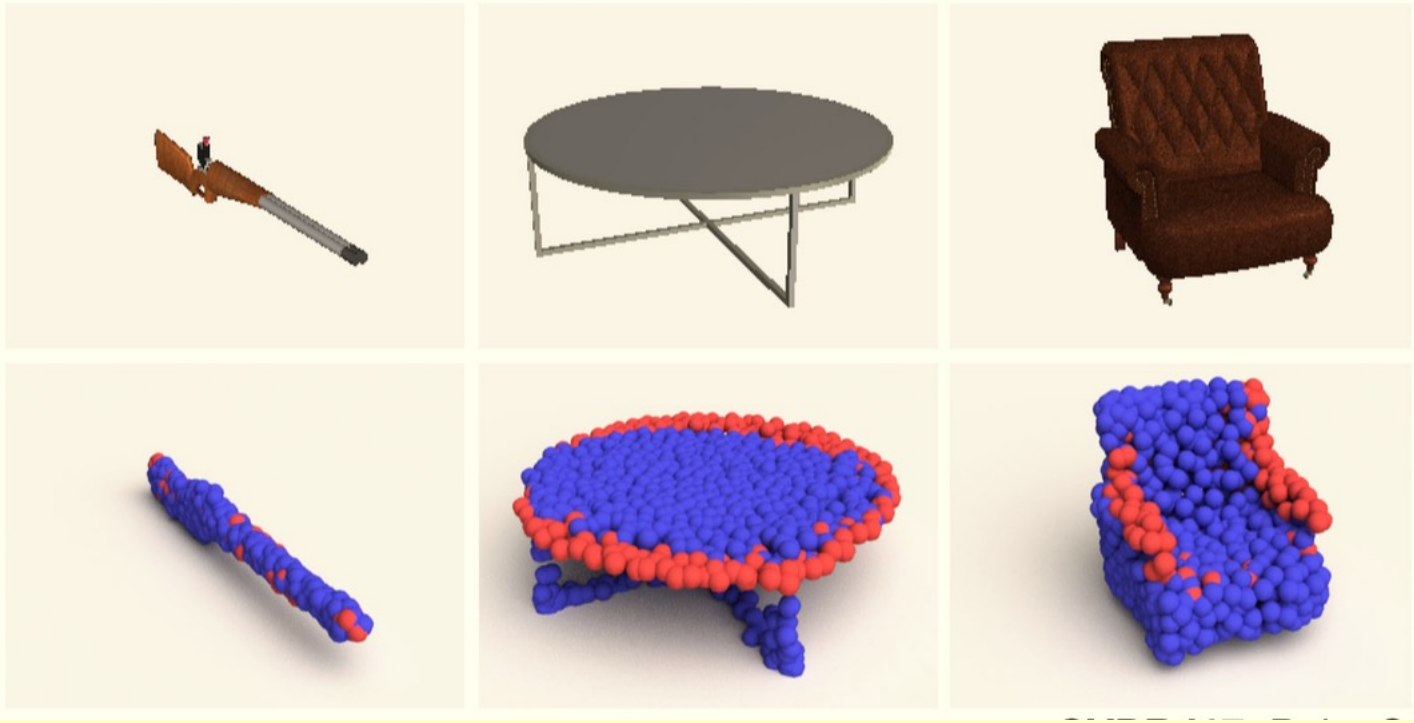
Point Set

Distance

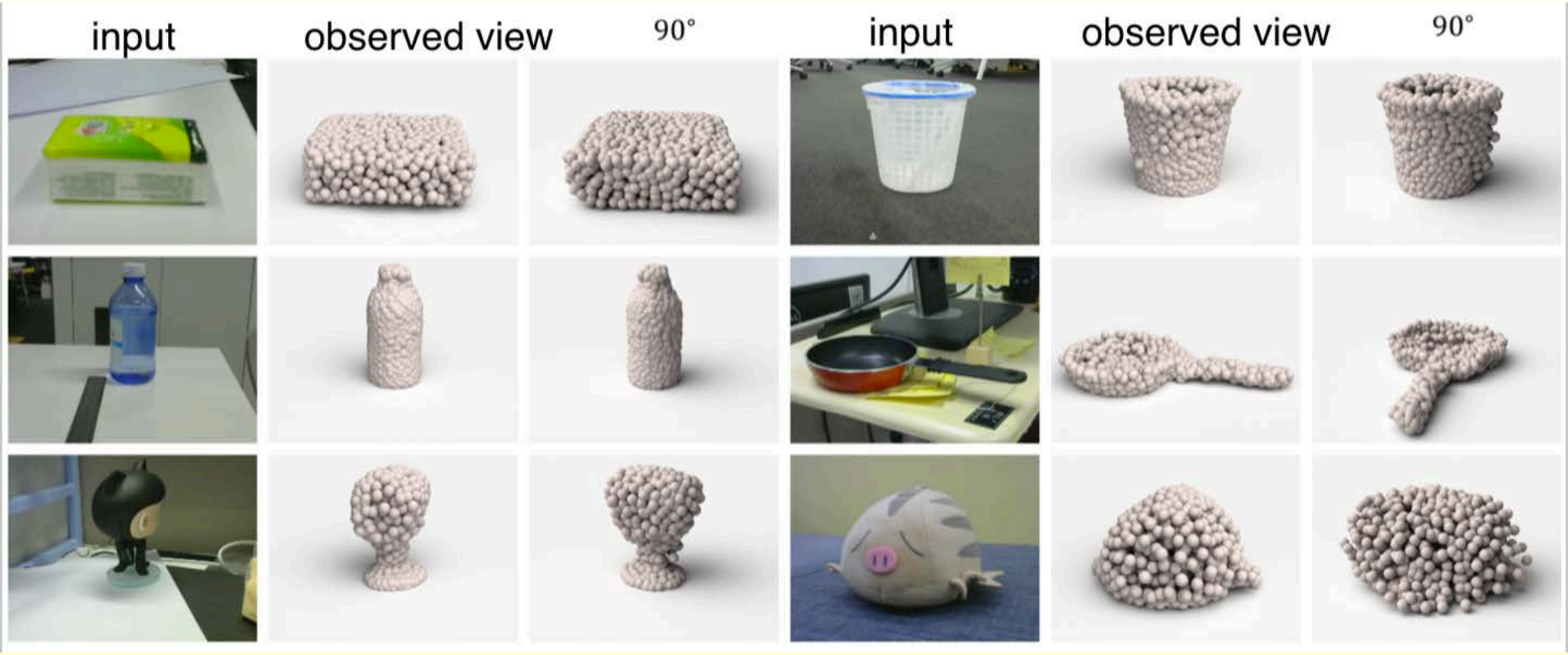
Which color corresponds to the deconv branch? FC branch?

blue: deconv branch – **large, smooth** structures

red: FC branch – **intricate** structures



Real world results



Conclusion and Outlook

- Deep learning for point cloud processing
  - PointNet, PointNet++
  - Application to 3D object detection
- Generating point clouds with deep networks

Future directions

- Robotics: Point cloud deep learning for object recognition, tracking, grasping, navigation etc.
- Shape design: Machine learning for shape design (e.g. AI-assisted shape editing – ComplementMe by Sung et al.)
- Point clouds are prevalent.. many opportunities!

References

[1] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *CVPR 2017*

[2] Charles R. Qi, Li Yi, Hao Su, and Leonidas Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *NIPS 2017*

[3] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds, *Arxiv.*

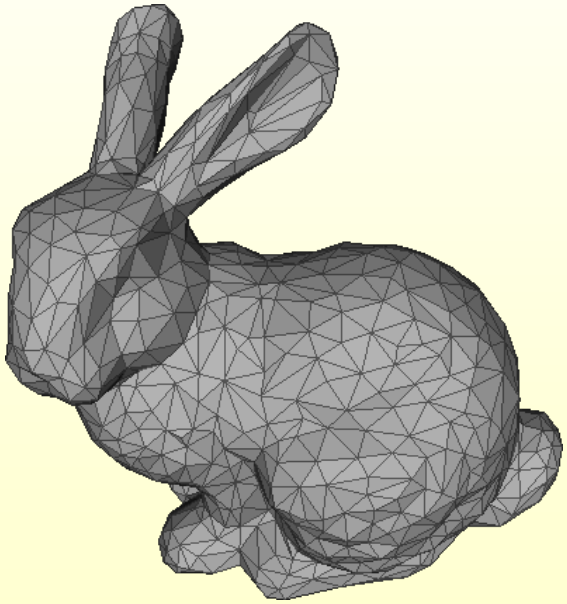
[4] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data, *CVPR 2018*

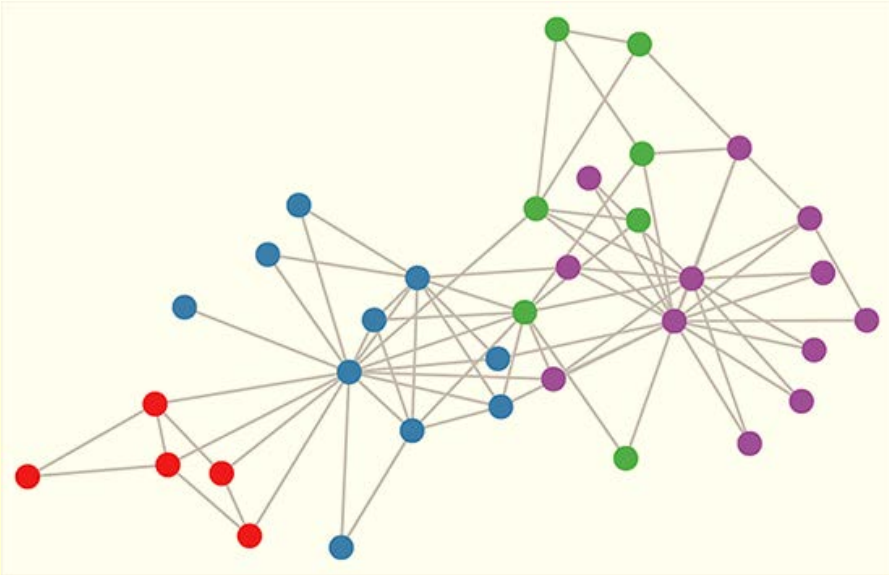
[5] Haoqiang Fan, Hao Su, and Leonidas Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image, *CVPR 2017*

Open Sourced Code

- PointNet: https://github.com/charlesq34/pointnet
- PointNet++: http://stanford.edu/~rqi/pointnet2/
- Dynamic Graph CNN: https://github.com/WangYueFt/dgcnn
- Frustum PointNets: https://github.com/charlesq34/frustum-pointnets
- Point Set Generation Net: https://github.com/fanhqme/PointSetGeneration

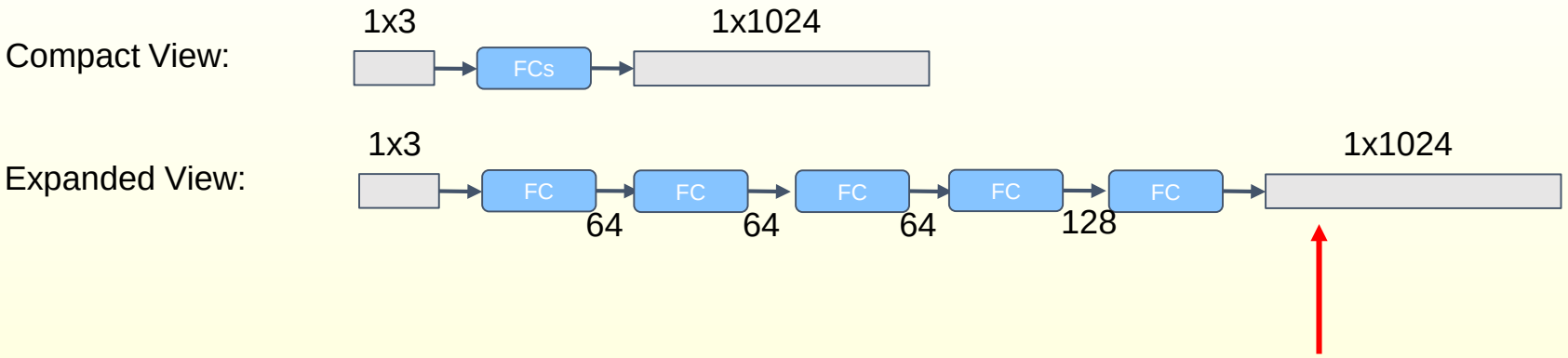
Next Lecture: Deep nets for meshes and graphs!





Backup Slides

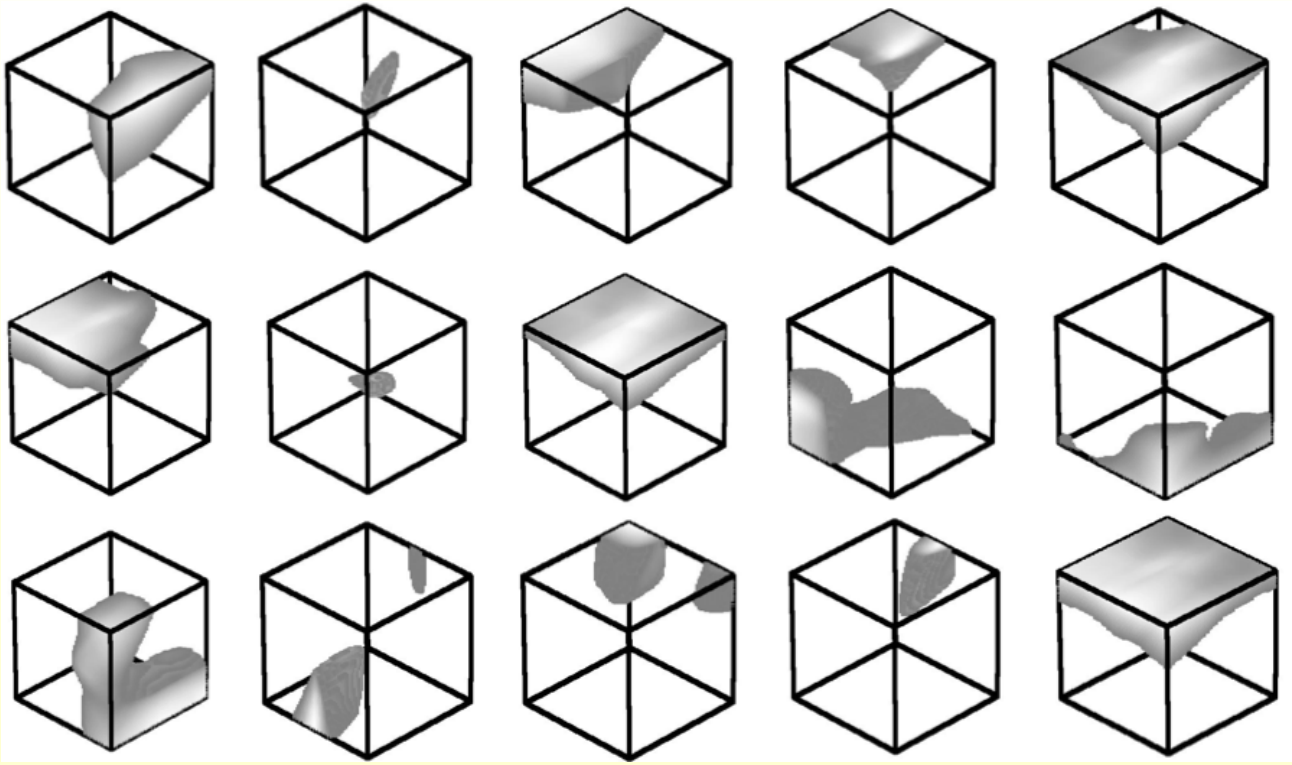
Visualizing Point Functions



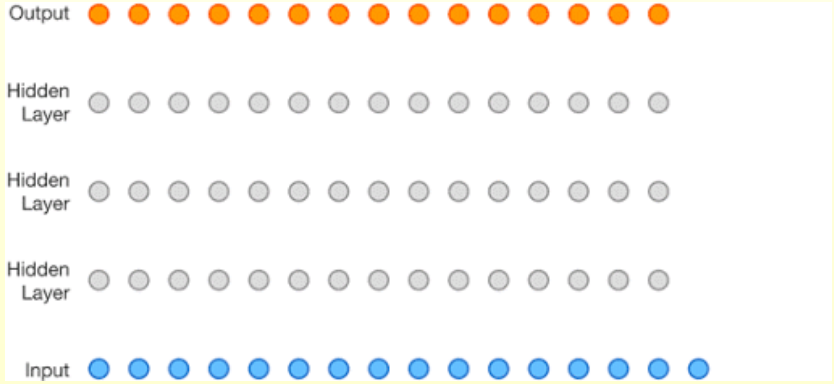
Which input point will activate neuron X?

Find the top-K points in a dense volumetric grid that activates neuron X.

Visualizing Point Functions



Acoustic Modeling: Deep Representation Learning



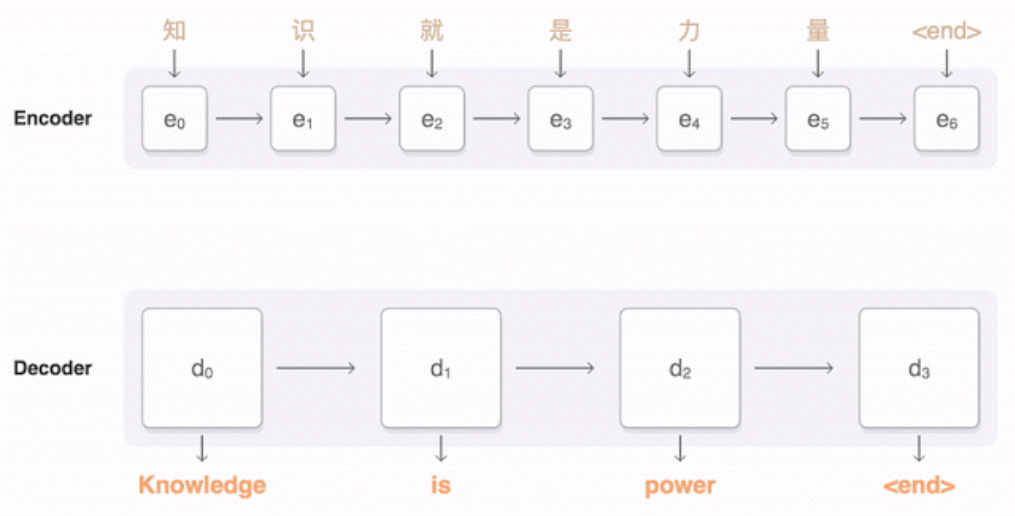
Acoustic Modeling

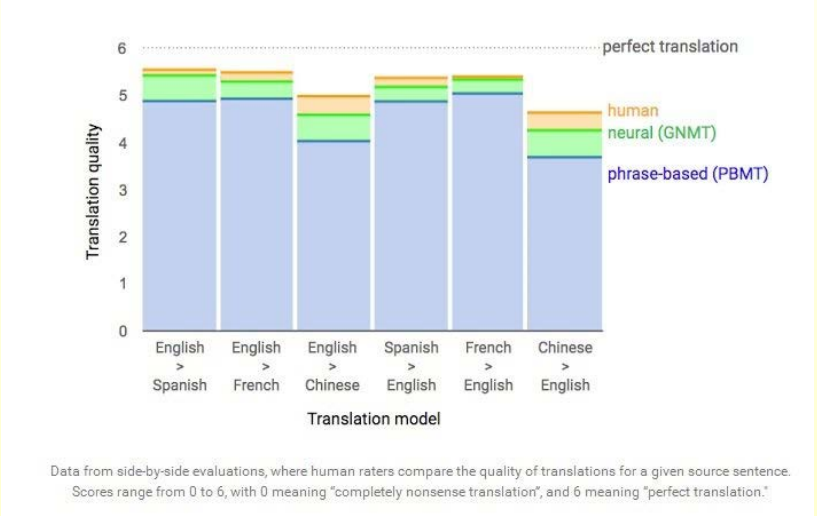
Near human-level Text-To-Speech performance

https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Machine translation:

Neural Translation Machine
by Quac V. Le et al at Google Brain.

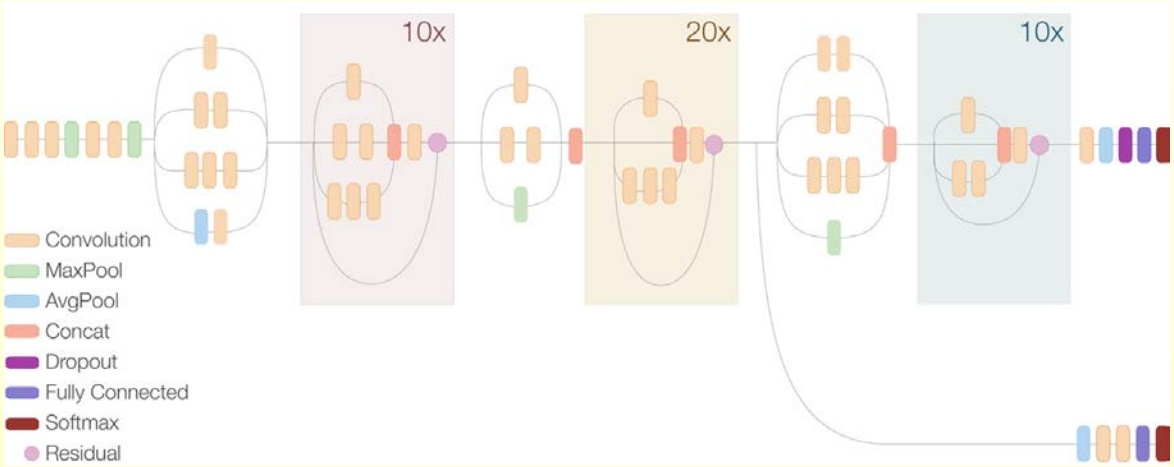




https://research.googleblog.com/2016/09/a-neural-network-for-machine.html

Big Data & Deep Representation Learning

Inception-ResNet-v2, with more than 150 layers
by Alex Alemi et al.



https://research.googleblog.com/2016/08/improving-inception-and-image.html

Image Classification

ImageNet 1000 class image classification:
Top-1 Accuracy: 80.4%
Top-5 Accuracy: 95.3%

Previous Work: Hand Crafted Point Cloud Features

Most existing point cloud features are **handcrafted** towards specific tasks

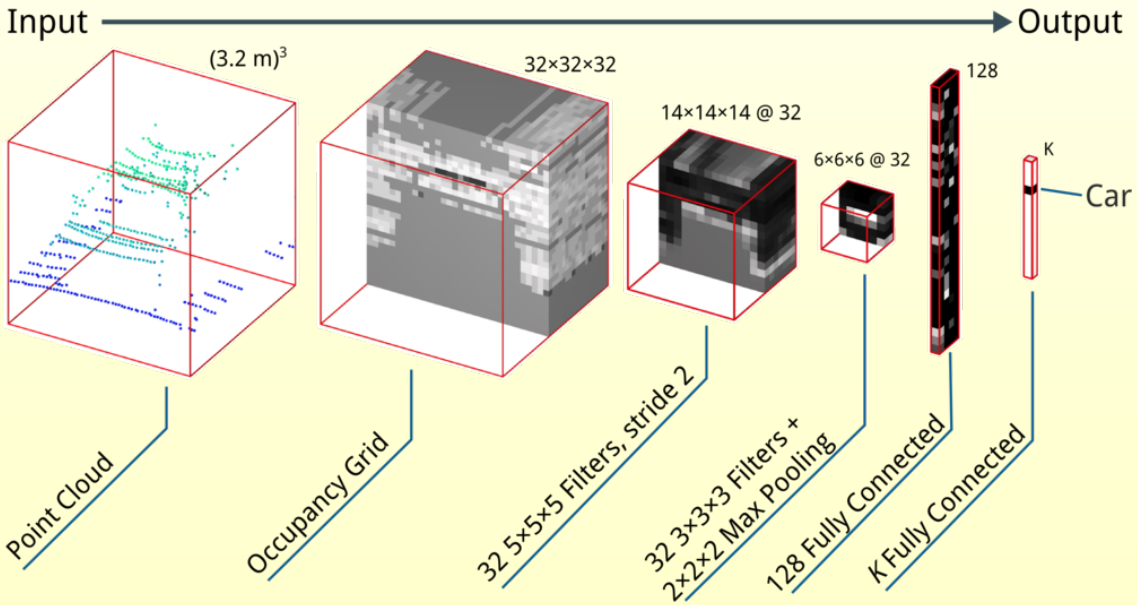
Feature Name	Supports Texture / Color	Local / Global / Regional	Best Use Case
PFH	No	L	
FPFH	No	L	2.5D Scans (Pseudo single position range images)
VFH	No	G	Object detection with basic pose estimation
CVFH	No	R	Object detection with basic pose estimation, detection of partial objects
RIFT	Yes	L	Real world 3D-Scans with no mirror effects. RIFT is vulnerable against flipping.

Source: https://github.com/PointCloudLibrary/pcl/wiki/Overview-and-Comparison-of-Features

Previous Work: Deep Network for Point Clouds

Point clouds converted to **volumetric grids**

- expensive 3d convolutions
- grid resolution?

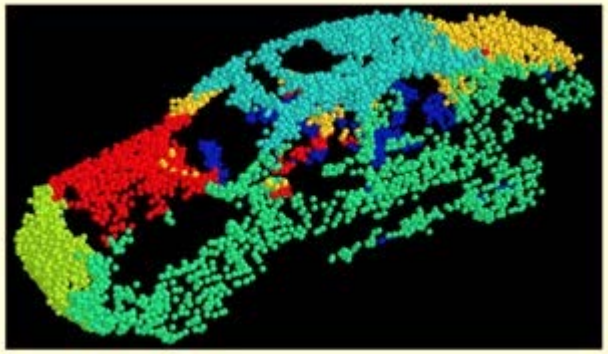


D. Maturana and S. Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In IROS. 2015

Previous Work: Deep Network for Point Clouds

Point clouds projected to **images**

- loss of 3d geometry
- perspective effect



Zelener, Allan, Philippos Mordohai, and Ioannis Stamos. "Classification of vehicle parts in unstructured 3d point clouds." 3DV 2014

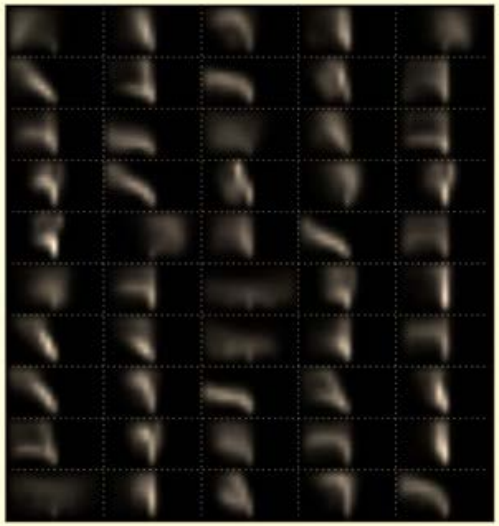
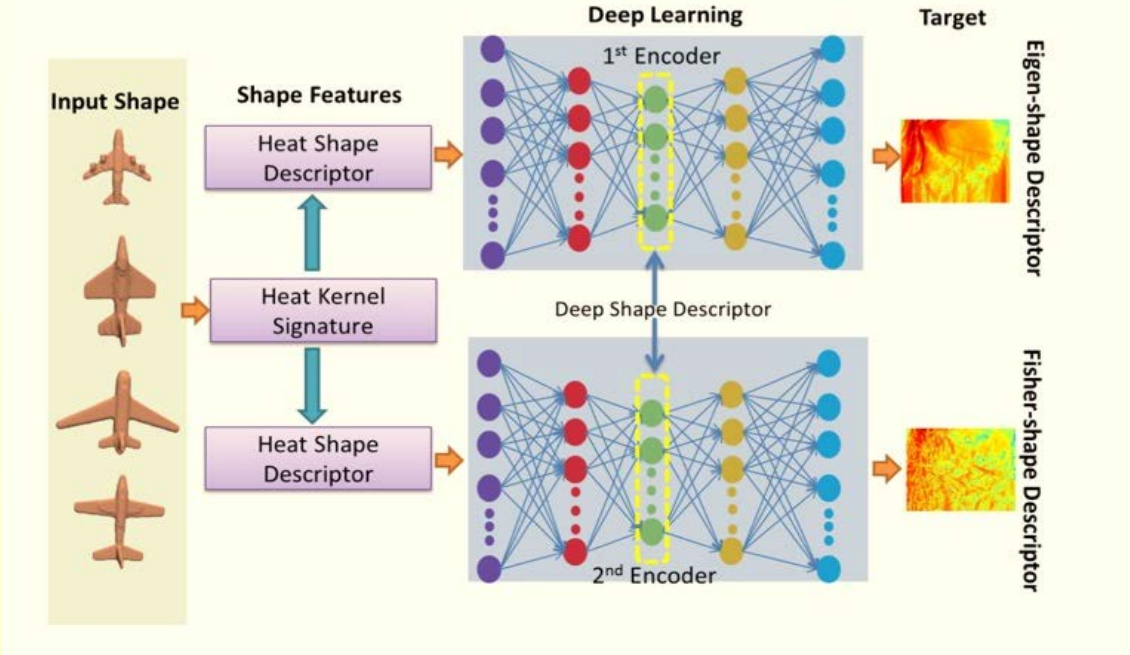


Figure 4. Spin image codebook. The bottom center of each spin image corresponds to the origin from which the spin image is computed. The y-axis corresponds to the radial direction and the x-axis to the cylinder height.

Previous Work: Deep Network for Point Clouds

Point clouds converted to **feature vectors**

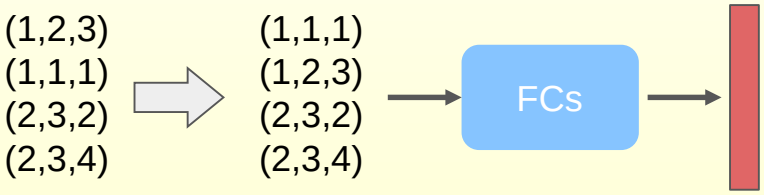


3D Deep Shape Descriptor by Y. Fang et al. CVPR15

Why not sort the input?

However, there is no canonical order in high dim space..

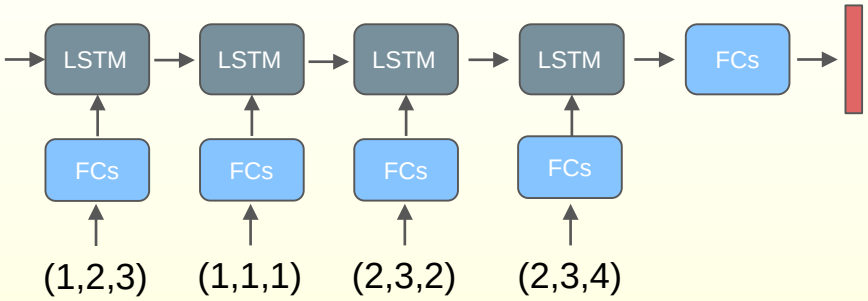
Use lexsort, sort by multiple keys, 1st dim then 2nd dim, 3rd dim



Multi-Layer Perceptron (ModelNet 40-class shape classification)

	Accuracy
Unordered Input	12%
Sorted Input	40%
PointNet (vanilla)	87%

How about RNNs?



Order Matters!
Order Matters: Sequence to Sequence for sets by Oriol Vinyals et al.

LSTM Network
(ModelNet shape 40-class classification)

	Accuracy
LSTM	75%
OrderMatterNet	75%
PointNet (vanilla)	87%

Challenges

Unordered point set as input

Model needs to be invariant to N! permutations.

Invariance under transformations

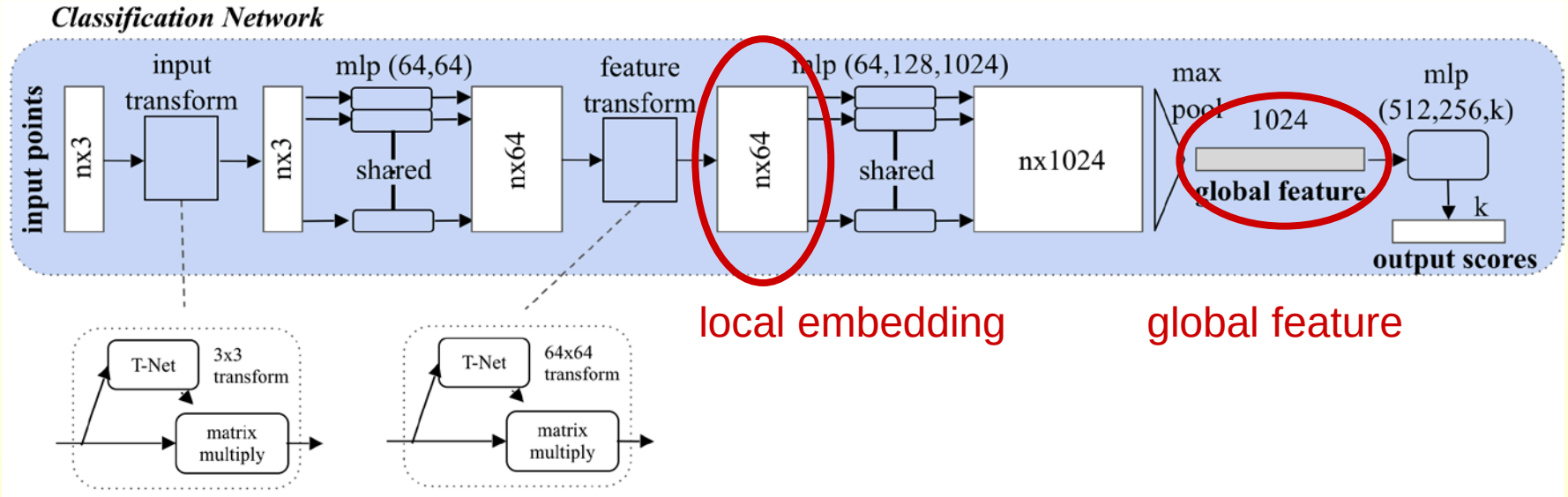
Point cloud rotations should not alter classification results.

Local and global info aggregation

Segmentation needs both local and global in

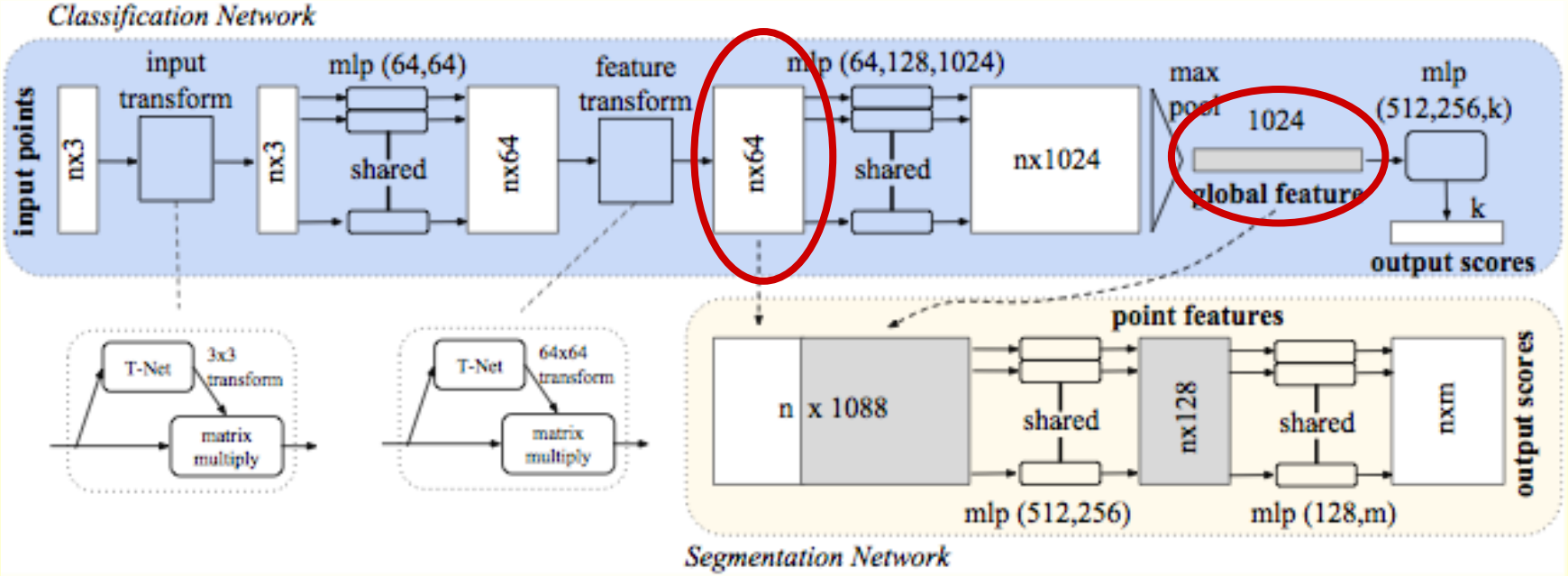
PointNet Segmentation Network

A simple approach: local embedding and global feature conca



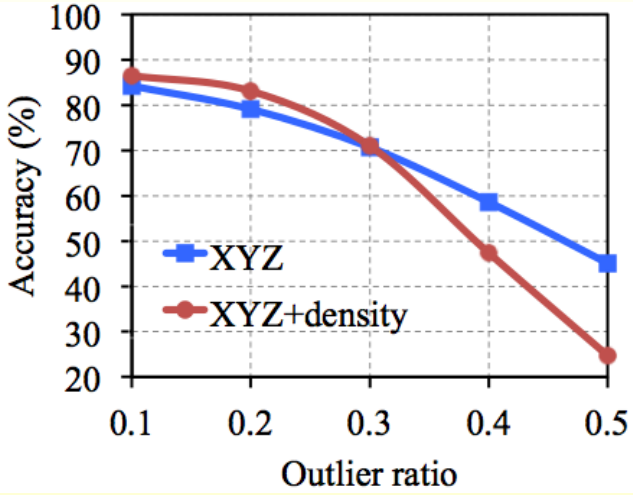
PointNet Segmentation Network

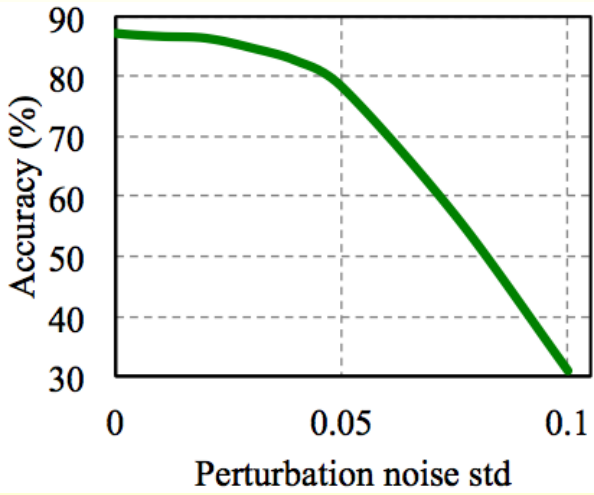
A simple approach: local embedding and global feature conca



Advanced and more scalable approach: PointNet++

Robustness to Data Corruption





dataset: ModelNet40; metric: 40-class classification accuracy (%)