Deep Learning for 3D Data

Dr. Surya Prakash
Associate Professor
Discipline of Computer Science & Engineering
Indian Institute of Technology Indore, Indore-453552, INDIA
E-mail: surya@iiti.ac.in

Application Areas

- **PointNet**
  - A deep learning framework for 3D point cloud data
- **Applications**
  - Object Classification
  - Object Recognition
    - Biometric Recognition
  - Object Part Segmentation
  - Semantic Scene Parsing

→ *R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "**PointNet**: Deep Learning on Point Sets for 3D Classification and Segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 77-85, DOI: 10.1109/CVPR.2017.16.*
→ ***Slide credit:** R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas (presentation at CVPR 2017)*

Outline

- **Overview of 3D deep learning**
  - *Why 3D?*
- 3D deep learning tasks
- 3D deep learning algorithms

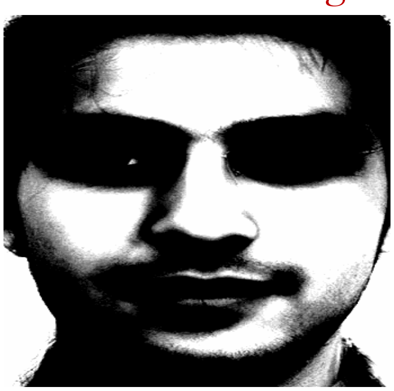
Overview of 3D Deep Learning

Introduction – *Why 3D*?

- The world around us is comprised of 3D geometry (objects)

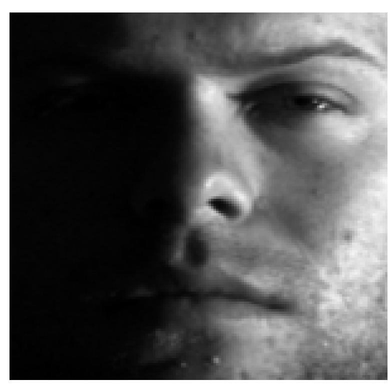
Introduction – *Why 3D*?

- Use of 2D images to represent 3D world - *issues*



Very high contrast

Very low contrast



Bad Illumination

Issues with respect to Image Quality

Introduction – *Why 3D*?

- Use of 2D images to represent 3D world - *issues*

Images of FERET database: different poses

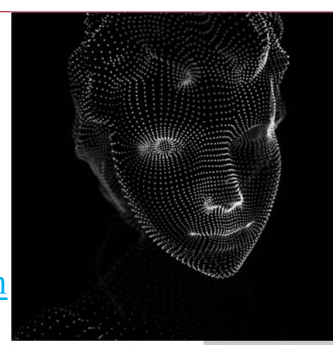
Rotational Issues

Image source: *Li, K., Huang, Q. Cross-pose face recognition by integrating regression iteration and interactive subspace. J Wireless Com Network* ***2019,*** *105 (2019). https://doi.org/10.1186/s13638-019-1429-*

7

Introduction – *Why 3D*?

- 3D data contains (x, y, z) values (can have color/texture as well)
- Insensitive to imaging problems such as lighting and shadows
- Can provide geometric information of the object
- 3D data can handle more general pose variations



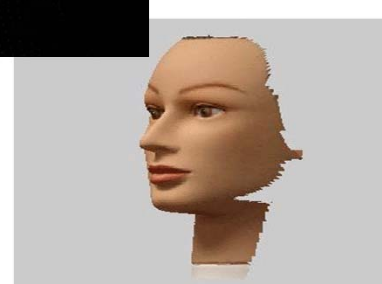
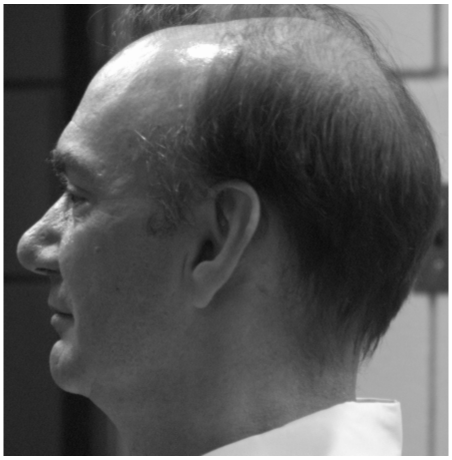
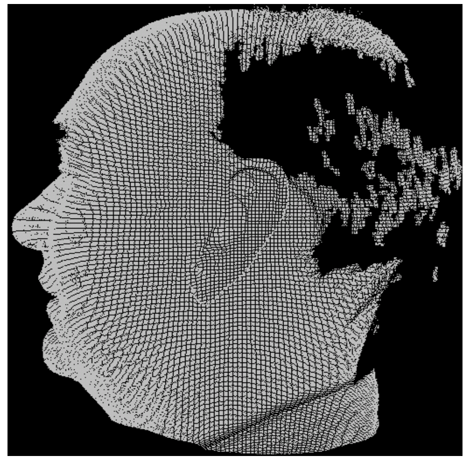


Image source: *http://cool-3d-pictures.blogspot.com/2010/10/3d-face.html*
https://in.pinterest.com/pin/519321400778202577/

Examples of 2D and 3D Data



2D Image



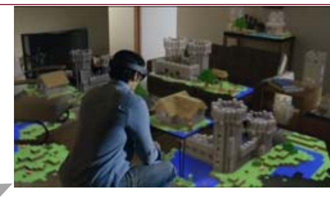
3D Image

Broad applications of 3D Data

Robotics

Broad applications of 3D data

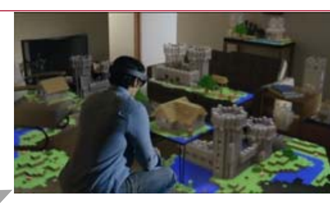
Robotics



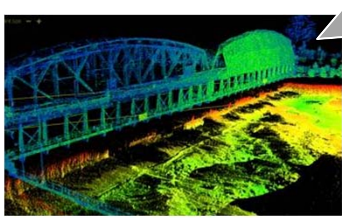
Augmented Reality

Broad applications of 3D data

Robotics



Augmented Reality

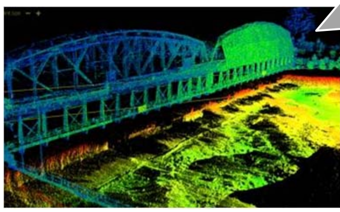


Autonomous driving

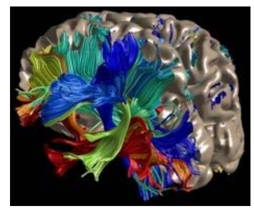
Broad Applications of 3D Data

Robotics

Augmented Reality



Autonomous driving



Medical Image Processing

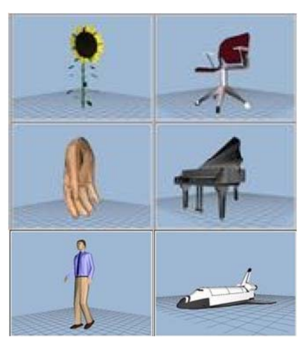
Lacking 3D data has been the major bottleneck

Status as of 2010:

Stanford bunny

Utah teapot

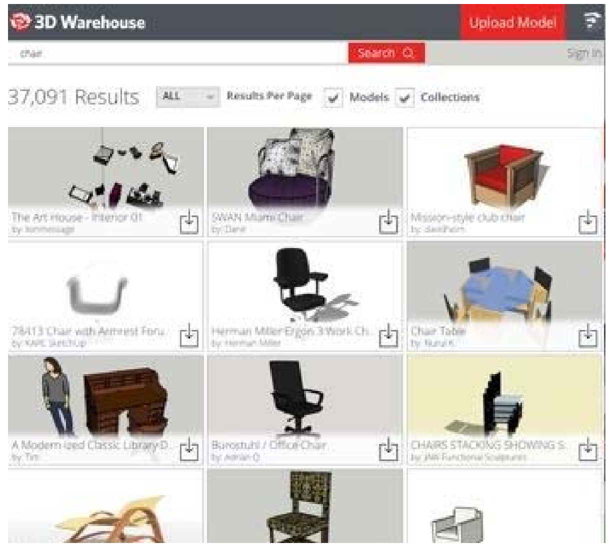
1800 models in **90** categories



Princeton shape benchmark
[Shilane et al. 04]

Recent Rise of Internet 3D models

- **Nowadays millions of 3D models in online repositories**



Recent Rise of Internet 3D models

- **Growing market of crowd-sourcing for 3D modeling**

yobi 3D

3D model search engine

Clara.io

Google SketchUp

Sketchfab

TINKERCAD

Recent rise of Internet 3D models

- **Growing market of crowd-sourcing for 3D modeling**

yobi3D Clara.io

An opportunity of Data-driven 3D Visual Computing

TINKERCAD

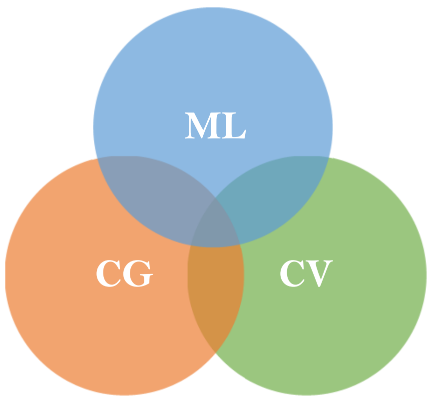
intel REALSENSE TECHNOLOGY

The Kinect sensor captures depth and color images simultaneously at a frame rate of up to 30 fps.

The integration of depth and color data results in a colored point cloud that contains about 300,000 points in every frame.

The Surge of 3D Deep Learning

- Arguably started from **2015** along with of big 3D datasets (ShapeNet & ModelNet)
- Very active due to huge industry interests!



- Robotics
- Autonomous driving
- Virtual/augmented reality
- Smart manufacturing
- …

Outline

- Overview of 3D deep learning
- Deep Learning Tasks
- 3D Deep Learning algorithms
  - 3D Representation issues
  - Deep learning on different 3D representations
    - Deep learning on regular structures
    - Deep learning on meshes
    - *Deep learning on point cloud and parametric models*

3D Deep Learning Tasks

3D Deep Learning Tasks

- **3D geometry analysis**
- **3D synthesis**
- **3D-assisted image analysis**

3D Deep Learning Tasks

- **3D geometry analysis**

Classification

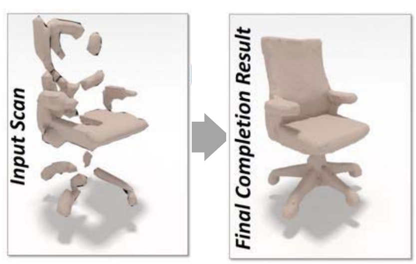
Parsing
(object/scene)

Correspondence

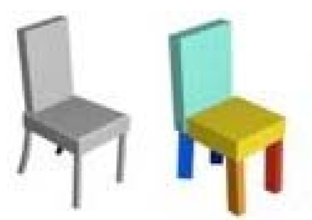
3D Deep Learning Tasks

- **3D synthesis**

Monocular
3D reconstruction



Shape completion



Shape modeling

3D Deep Learning Tasks

- **3D-assisted image analysis**

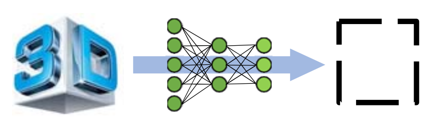
Cross-view image retrieval

Intrinsic decomposition

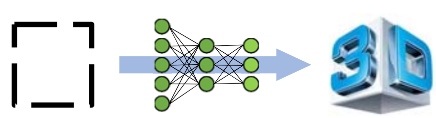
Process of separating an image into its formation components such as **reflectance** (color) and **shading** (illumination).

All about **Data** and **Network**

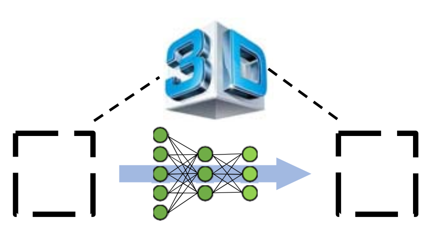
- **3D geometry analysis**



- **3D synthesis**

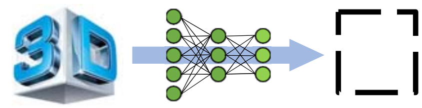


- **3D-assisted image analysis**

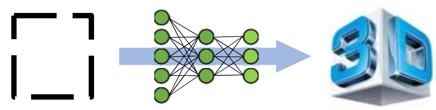


All about **Data** and **Network**

- **3D geometry analysis**



- **3D synthesis**



Outline

- Overview of 3D deep learning
- Deep Learning Tasks
- **3D Deep Learning algorithms**
  - **3D Representation issues**
  - Deep learning on different 3D representations
    - Deep learning on regular structures
    - Deep learning on meshes
    - Deep learning on point cloud and parametric models

3D Deep Learning algorithms

3D Representation issues

3D Representation Issues

- Images: Unique representation with regular data structure

1	44	33	12	20	23	35	14
51	16	40	32	46	48	28	17
29	60	3	63	49	55	36	7
52	22	26	41	38	10	61	53
2	24	19	11	34	43	5	8
57	9	37	42	25	21	27	18
30	56	50	64	4	59	6	13
58	47	45	31	39	15	62	54

The Representation Issues of 3D Deep Learning

3D has many representations:

multi-view RGB(D) images

volumetric

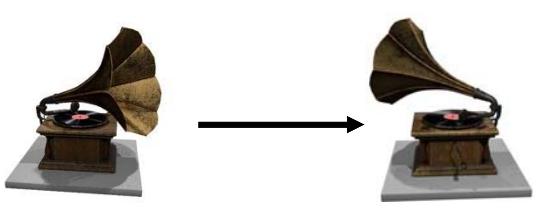
polygonal mesh

point cloud

primitive-based CAD models

The Representation Issues of 3D Deep Learning

3D has many representations:



Novel view image synthesis

multi-view RGB(D) images

volumetric

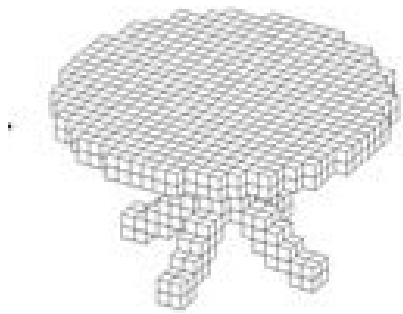
Depth image

polygonal mesh

point cloud

primitive-based CAD models

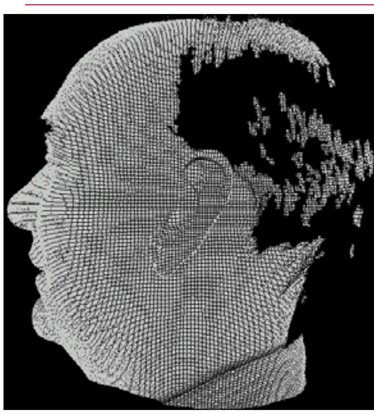
The Representation Issues of 3D Deep Learning



3D has many representations:

- multi-view RGB(D) images
- **volumetric**
- Depth image
- polygonal mesh
- point cloud
- primitive-based CAD models

The Representation Issues of 3D Deep Learning



3D has many representations:

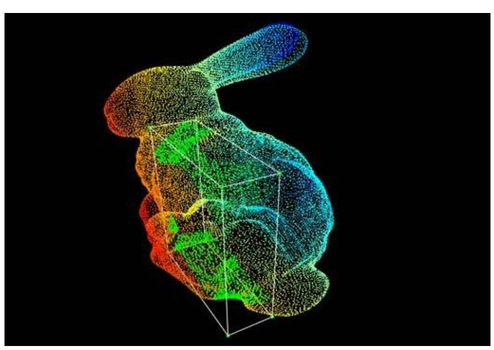
- multi-view RGB(D) images
- volumetric
- **Depth image**
- polygonal mesh
- point cloud
- primitive-based CAD models

The Representation Issues of 3D Deep Learning

3D has many representations:

- multi-view RGB(D) images
- volumetric
- Depth image
- **polygonal mesh**
- point cloud
- primitive-based CAD models

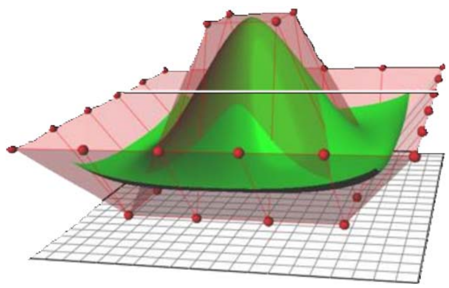
The Representation Issues of 3D Deep Learning



3D has many representations:

- multi-view RGB(D) images
- volumetric
- Depth image
- polygonal mesh
- **point cloud**
- primitive-based CAD models

The Representation Issues of 3D Deep Learning



3D has many representations:

multi-view RGB(D) images

volumetric

Depth image

polygonal mesh

point cloud

primitive-based CAD models

The Representation Issues of 3D Deep Learning

- 3D has many representations:

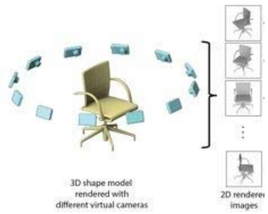
Rasterized form (regular grids)

- multi-view RGB(D) images
- volumetric
- depth images

Geometric form (irregular)

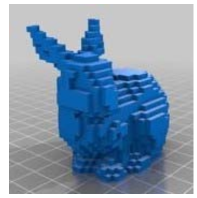
- polygonal mesh
- point cloud
- primitive-based CAD models

3D Deep Learning Algorithms (By Representations)



Multi-view

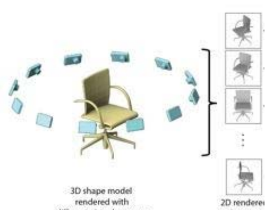
[Su et al. 2015]
[Kalogerakis et al. 2016]
…



Volumetric

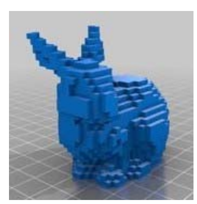
[Maturana et al. 2015]
[Wu et al. 2015] (GAN)
[Qi et al. 2016]
[Liu et al. 2016]
[Wang et al. 2017] (O-Net)
[Tatarchenko et al. 2017] (OGN)
…

3D Deep Learning Algorithms (By Representations)



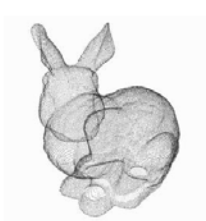
Multi-view

[Su et al. 2015]
[Kalogerakis et al. 2016]
…



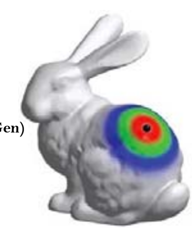
Volumetric

[Maturana et al. 2015]
[Wu et al. 2015] (GAN)
[Qi et al. 2016]
[Liu et al. 2016]
[Wang et al. 2017] (O-Net)
[Tatarchenko et al. 2017] (OGN)
…



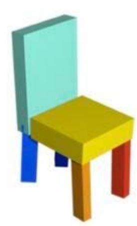
[Qi et al. 2017] (PointNet)
[Fan et al. 2017] (PointSetGen)

Point cloud



[Henaff et al. 2015]
[Defferard et al. 2016]
[Yi et al. 2017] (SyncSpecCNN)
…

Mesh (Graph CNN)

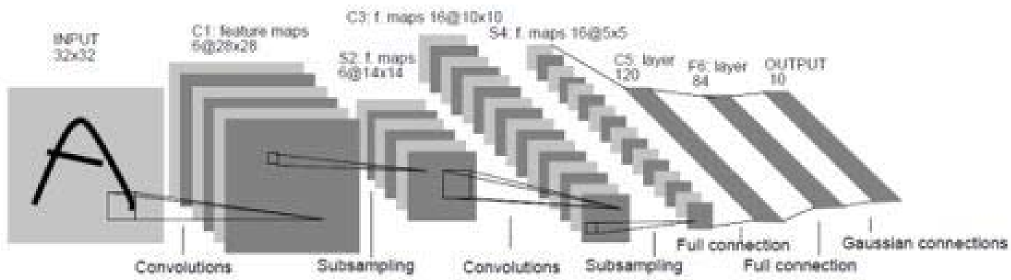


[Tulsiani et al. 2017]
[Li et al. 2017] (GRASS)

Part assembly

Fundamental Challenges of 3D Deep Learning

- Can we directly apply CNN on 3D data?



Fundamental Challenges of 3D Deep Learning

- Can we directly apply CNN on 3D data?

1	44	33	12	20	23	35	14
51	16	40	32	46	48	28	17
29	60	3	63	49	55	36	7
52	22	26	41	38	10	61	53
2	24	19	11	34	43	5	8
57	9	37	42	25	21	27	18
30	56	50	64	4	59	6	13
58	47	45	31	39	15	62	54

$$(f * g)[n] = \sum_{m=-M}^{M} f[n-m]g[m]$$

Fundamental Challenges of 3D Deep Learning

- 3D has many representations:
  - multi-view RGB(D) images
  - depth image
  - volumetric
  - polygonal mesh
  - point cloud
  - primitive-based CAD models

Rasterized form (regular grids)

- Can directly apply CNN
- But has other challenges

Fundamental Challenges of 3D Deep Learning

- 3D has many representations:
  - multi-view RGB(D) images
  - Volumetric
  - polygonal mesh
  - point cloud
  - primitive-based CAD models

Geometric form (irregular)

Cannot directly apply CNN

Outline

- Overview of 3D deep learning
- Deep Learning Tasks
- **3D Deep Learning algorithms**
  - 3D Representation issues
  - Deep learning on different 3D representations
    - **Deep learning on regular structures**
    - **Deep learning on meshes**
    - ***Deep learning on point cloud and parametric models***

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

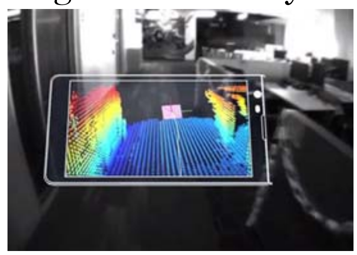
Motivation

- Big Data + Deep Representation Learning

Robot Perception

source: Scott J Grunewald

Augmented Reality



source: Google Tango

Shape Design

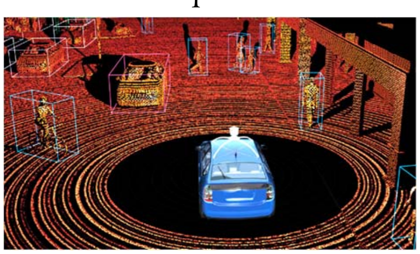
source: solidsolutions

Emerging 3D Applications

Motivation

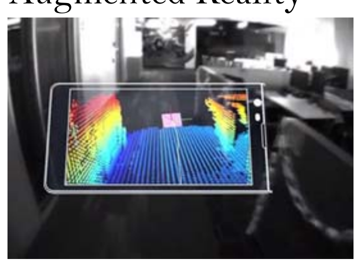
- Big Data + Deep Representation Learning

Robot Perception



source: Scott J Grunewald

Augmented Reality



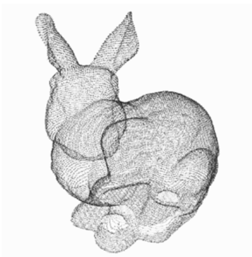
source: Google Tango

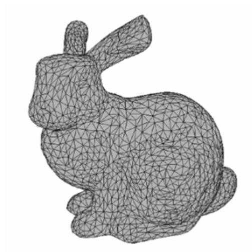
Shape Design

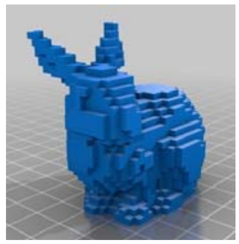
source: solidsolutions

Need for 3D Deep Learning!

3D Representations


Point Cloud


Mesh

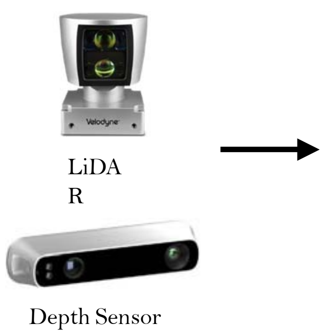

Volumetric

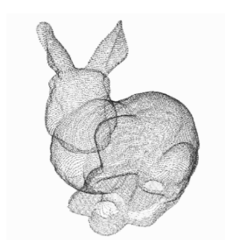

Projected View
RGB(D)

…

3D Representation: Point Cloud

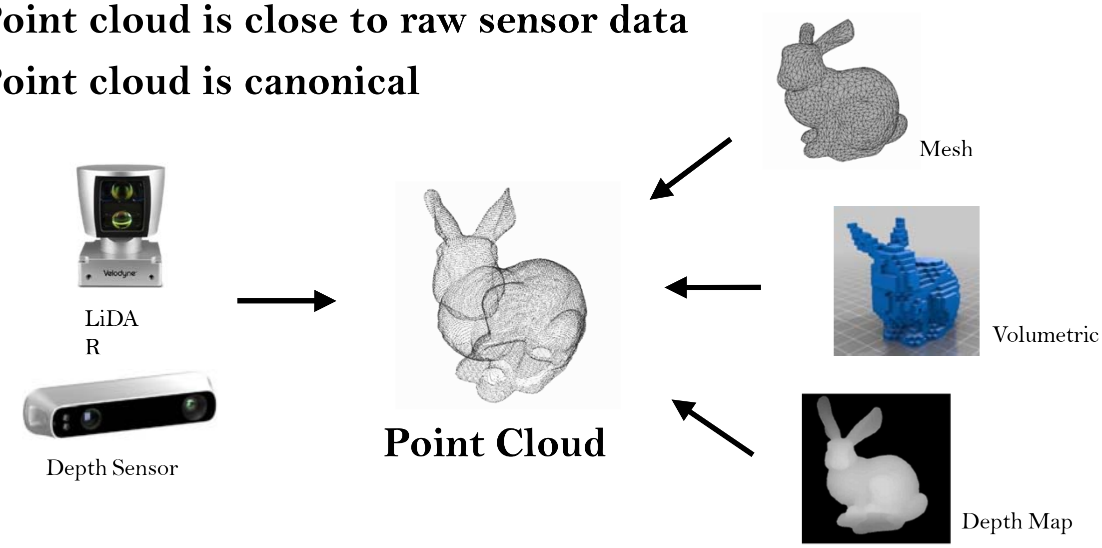
- **Point cloud is close to raw sensor data**




Point Cloud

3D Representation: Point Cloud

- **Point cloud is close to raw sensor data**
- **Point cloud is canonical**



Previous Works

- 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 Works – *Use of Deep Learning*

- Point cloud is an important type of geometric data structure.
- Due to its irregular format, most researchers transform such data to regular 3D voxel grids or collections of images.
- This renders data unnecessarily voluminous and causes issues.

Previous Works

- Point cloud is **converted to other representations** before it is fed to a deep neural network

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

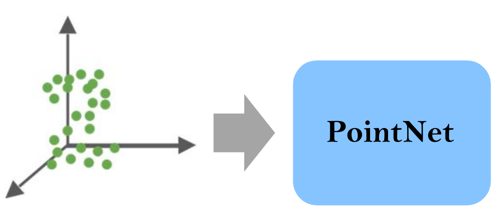
Research Question

- Can we achieve effective **feature learning directly on** point clouds?

- Yes, using PointNet
  - PointNet provides a neural network that directly consumes point clouds, which well respects the permutation invariance of points in the input.

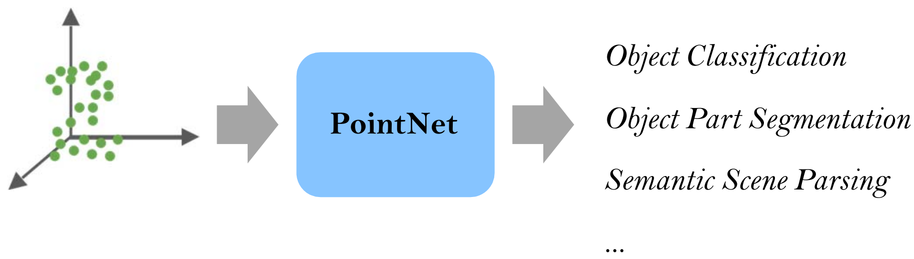
PointNet

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



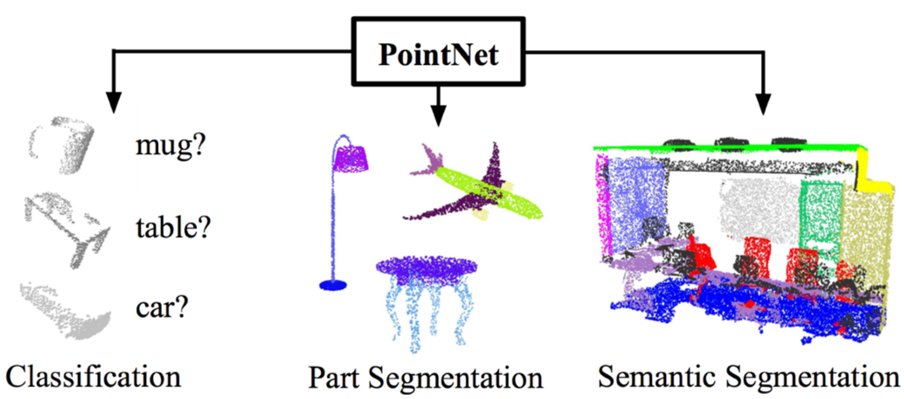
PointNet

- End-to-end learning for **scattered, unordered** point data
- **Unified** framework for various tasks



PointNet

- End-to-end learning for **scattered, unordered** point data
- **Unified** framework for various tasks



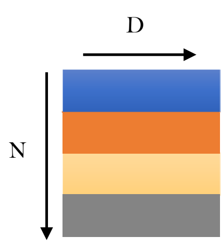
Challenges

- **Unordered point set as input**
  - **Model needs to be invariant to *N!* permutations.**

- Invariance under geometric transformations
  - Point cloud rotations should not alter classification results.

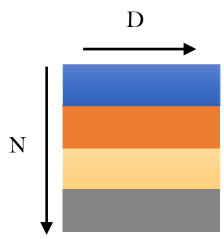
Unordered Input

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

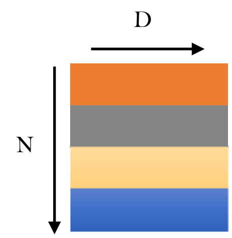


Unordered Input

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

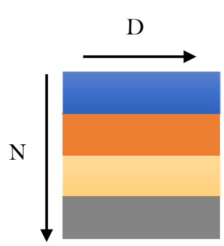


represents the same **set** as

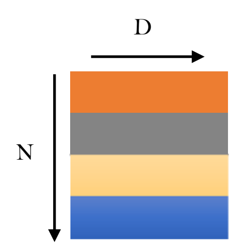


Unordered Input

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



represents the same **set** as



Model needs to be invariant to *N!* permutations

Unordered Input

File-1 (.asc)

1	80.993805	84.056732	38.899254
2	80.993805	83.522430	39.102684
3	81.182098	83.522430	38.899254
4	81.993805	85.522430	37.154747
5	82.201813	85.522430	36.899254
6	81.715118	84.522430	37.899254
7	81.993805	83.773407	37.899254
8	81.993805	83.522430	37.993004
9	81.993805	84.522430	37.582420
10	82.559235	84.522430	36.899254
11	82.910248	83.522430	36.899254
12	83.412994	88.522430	33.899254
13	83.725555	87.522430	33.899254
14	83.993805	88.673065	32.899254
15	83.993805	87.522430	33.492638
16	84.345428	87.522430	32.899254
17	82.993805	86.522430	35.447838

File-2 (.asc)

1	82.910248	83.522430	36.899254
2	83.412994	88.522430	33.899254
3	83.725555	87.522430	33.899254
4	83.993805	88.673065	32.899254
5	83.993805	87.522430	33.492638
6	84.345428	87.522430	32.899254
7	82.993805	86.522430	35.447838
8	83.361481	86.522430	34.899254
9	82.993805	85.487213	35.899254
10	83.681488	85.522430	34.899254
11	82.993805	84.522430	36.334862
12	83.317535	84.522430	35.899254
13	83.654144	83.522430	35.899254
14	83.993805	84.575714	34.899254
15	83.993805	83.522430	35.421043
16	84.337799	83.522430	34.899254
17	83.993805	86.628998	33.899254

Both the files represent the same face though the order of the points is different

First 10 points of the first .asc file have been appended at the end in the second .asc file.

Permutation Invariance: Symmetric Function

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

Permutation Invariance: Symmetric Function

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

Examples:

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

Permutation Invariance: Symmetric Function

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

Examples:

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

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

Permutation Invariance: Symmetric Function

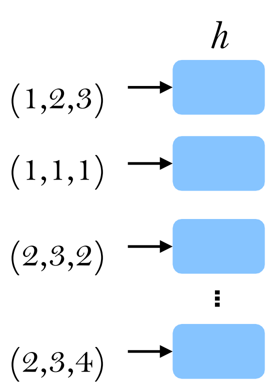
- **Observe:**

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

Permutation Invariance: Symmetric Function

- **Observe:**

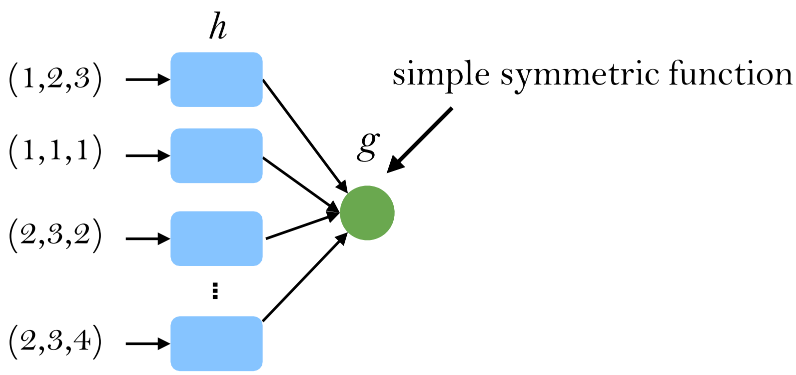
$f(x_1, x_2, \ldots, x_n) = \gamma \; o \; g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Permutation Invariance: Symmetric Function

- **Observe:**

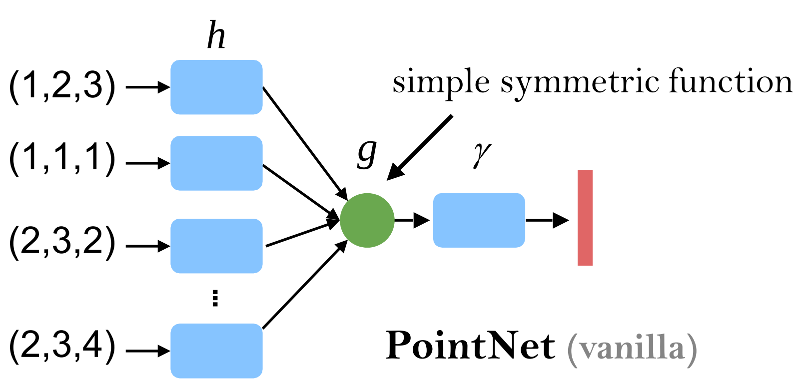
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Permutation Invariance: Symmetric Function

- **Observe:**

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



Permutation Invariance: Symmetric Function

Original space — Embedding in higher dimensional space

(1,2,3) → (**1, 2, 3,** 0, 0, 0, 0, 0, 0, 0, 0, 0)

(1,1,1) → (0, 0, 0, **1, 1, 1**, 0, 0, 0, 0, 0, 0)

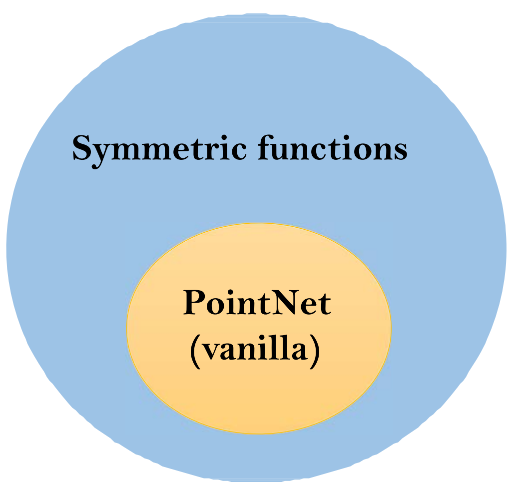
(2,3,2) → (0, 0, 0, 0, 0, 0 , **2, 3, 2**, 0, 0, 0)

(2,3,4) → (0, 0, 0, 0, 0, 0, 0, 0, 0, **2, 3, 4**)

(**1, 2, 3**, **1, 1, 1**, **2, 3, 2**, **2, 3, 4**) ← max

Permutation Invariance: Symmetric Function

- What symmetric functions can be constructed by PointNet?



Universal Set Function Approximator

- **Theorem:**
  - A <u>Hausdorff continuous symmetric function</u> $f : 2^{X} \rightarrow R$ can be arbitrarily approximated by PointNet.

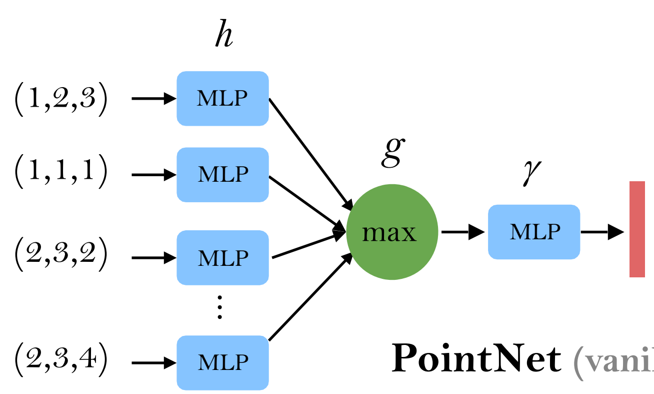
$$\left| f(S) - \boxed{\gamma \left(\underset{x_i \in S}{\mathrm{MAX}} \{h(x_i)\} \right)} \right| < \epsilon$$

$S \subseteq R^{d}$

PointNet (vanilla)

Basic PointNet Architecture

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



PointNet (vanilla)

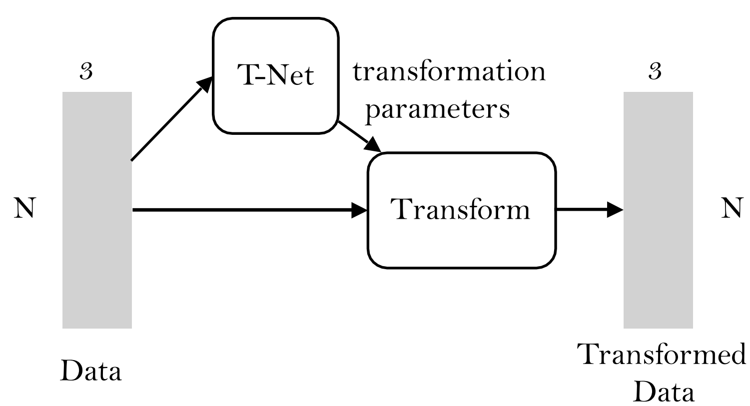
Challenges

- <u>Unordered</u> point set as input
  – Model needs to be invariant to *N!* permutations.

- **Invariance under <u>geometric transformations</u>**
  – Point cloud rotations should not alter classification results.

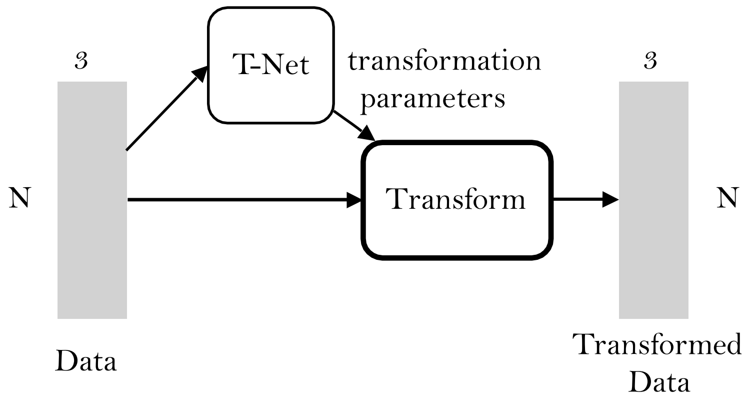
Input Alignment by Transformer Network

- Idea: Data dependent transformation for <u>automatic alignment</u>



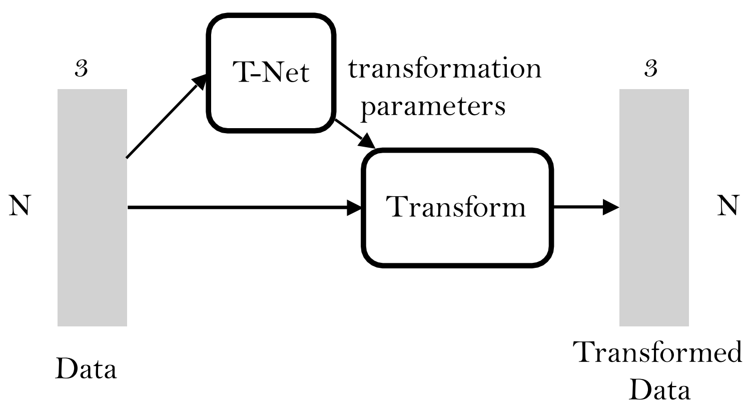
Input Alignment by Transformer Network

- Idea: Data dependent transformation for <u>automatic alignment</u>



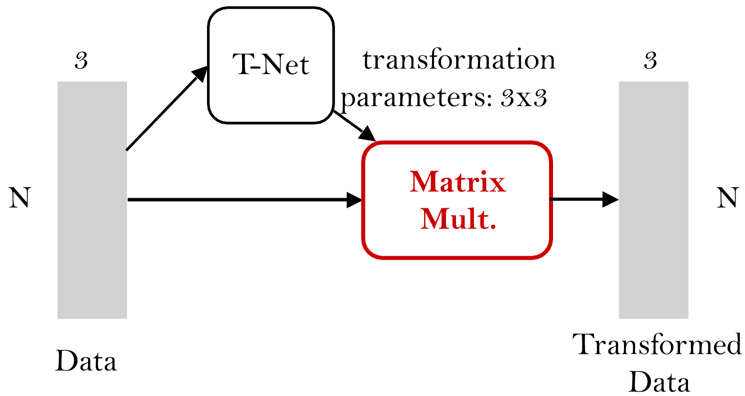
Input Alignment by Transformer Network

- Idea: Data dependent transformation for <u>automatic alignment</u>

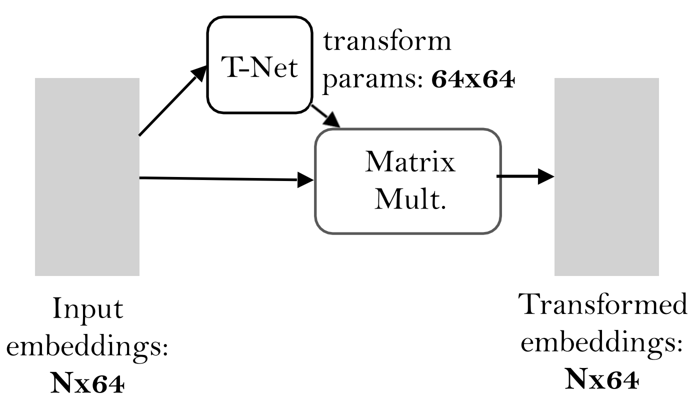


Input Alignment by Transformer Network

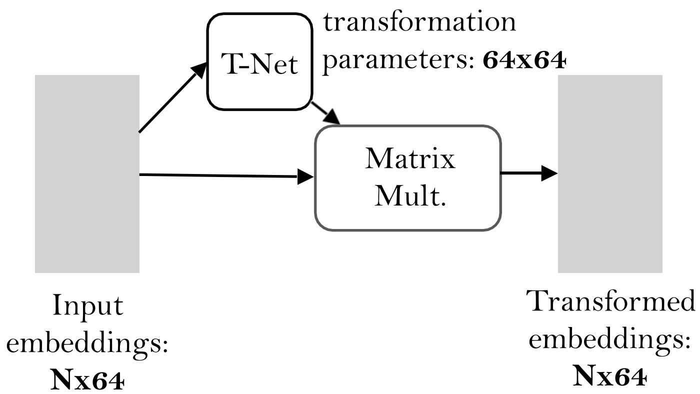
- The transformation is just matrix multiplication!



Embedding Space Alignment



Embedding Space Alignment

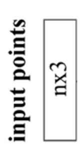


Regularization:

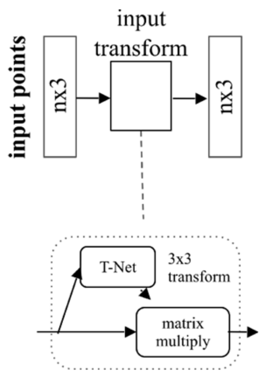
Transform matrix A 64x64 close to orthogonal:

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

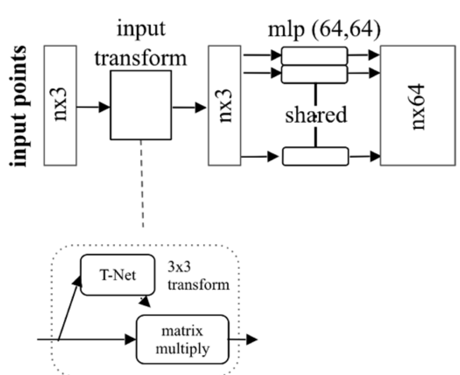
PointNet Classification Network



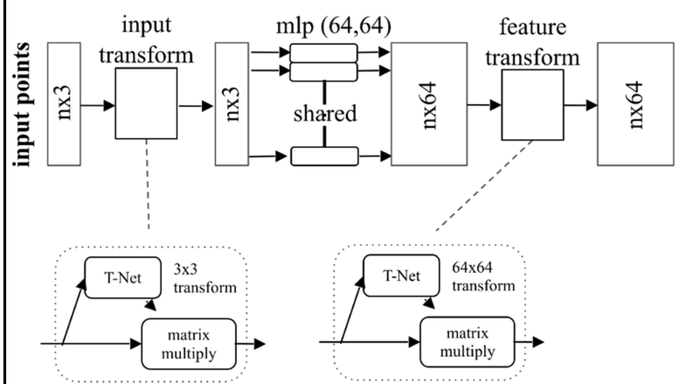
PointNet Classification Network



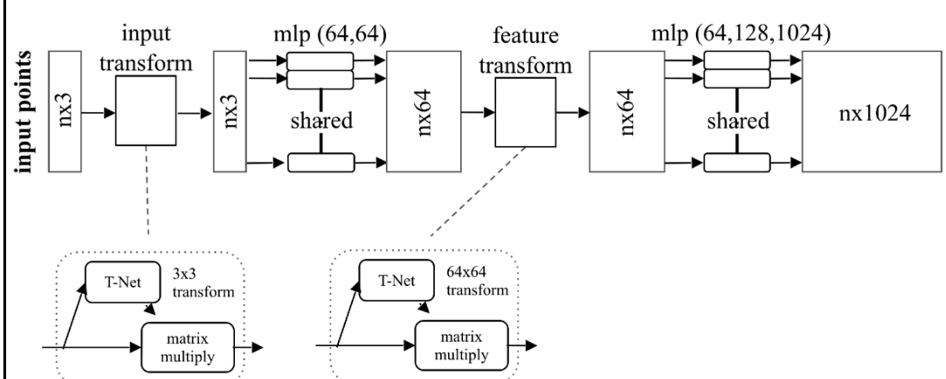
PointNet Classification Network



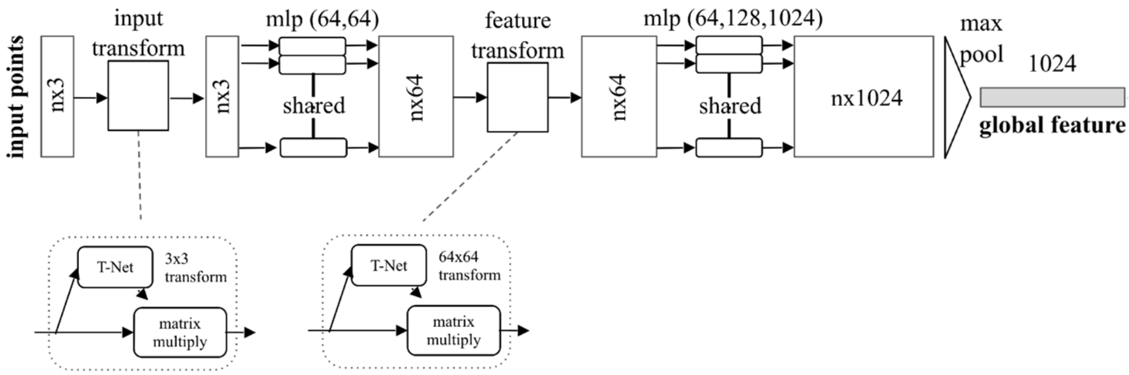
PointNet Classification Network



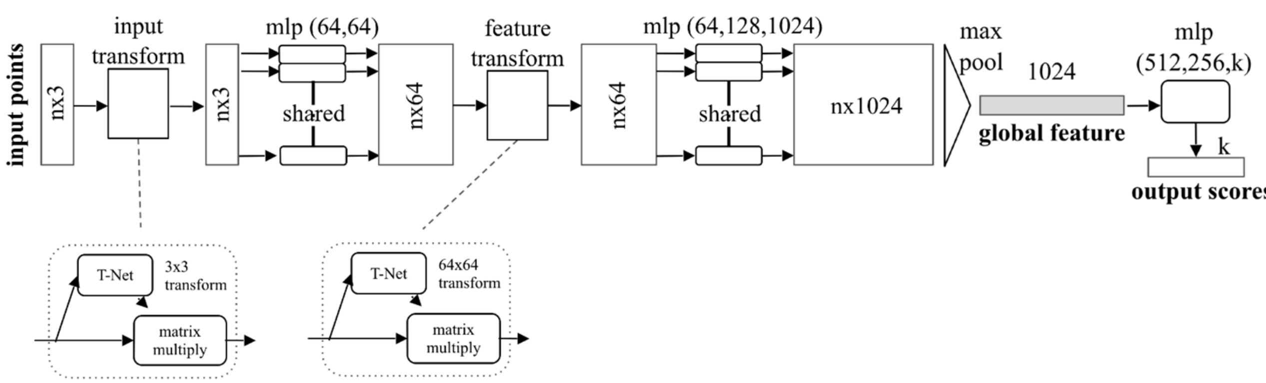
PointNet Classification Network



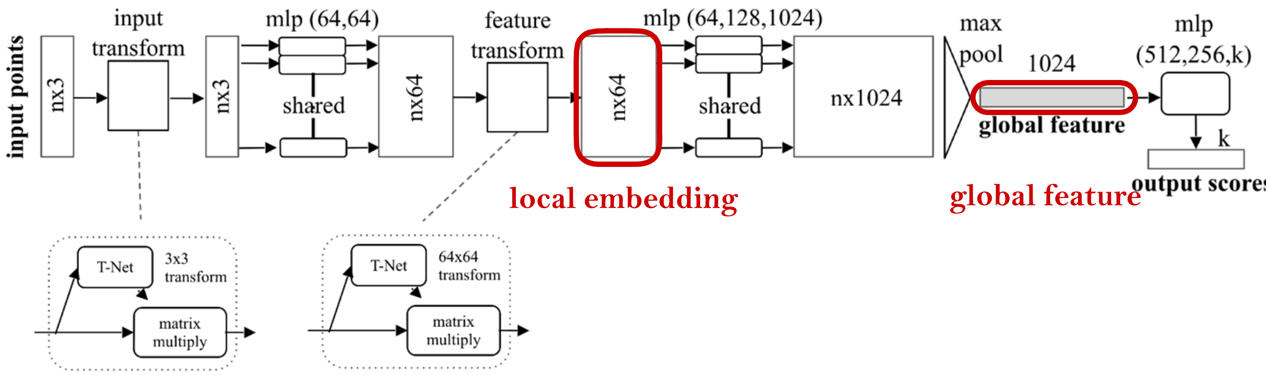
PointNet Classification Network



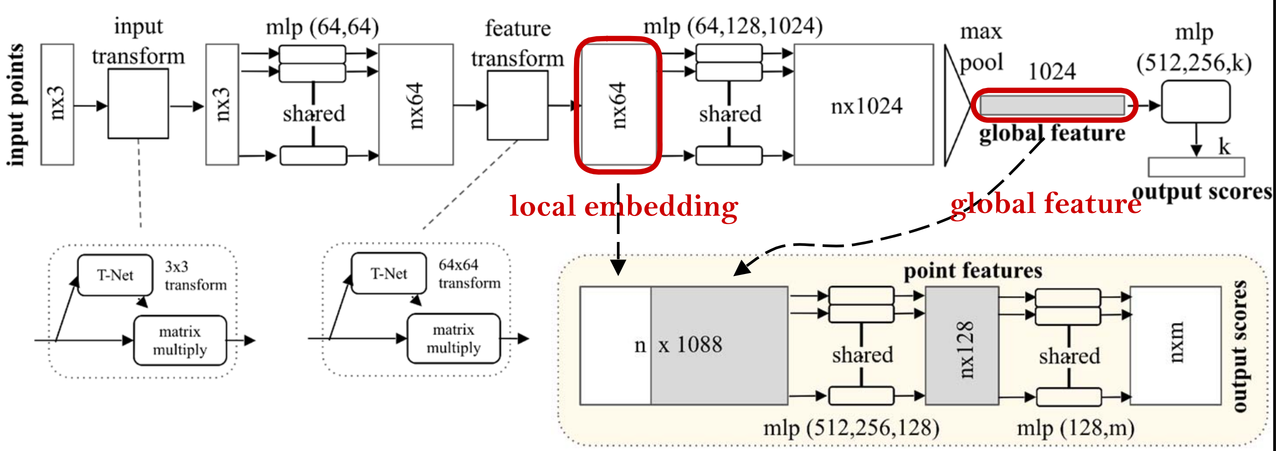
PointNet Classification Network



Extension to PointNet Segmentation Network



Extension to PointNet Segmentation Network



Results

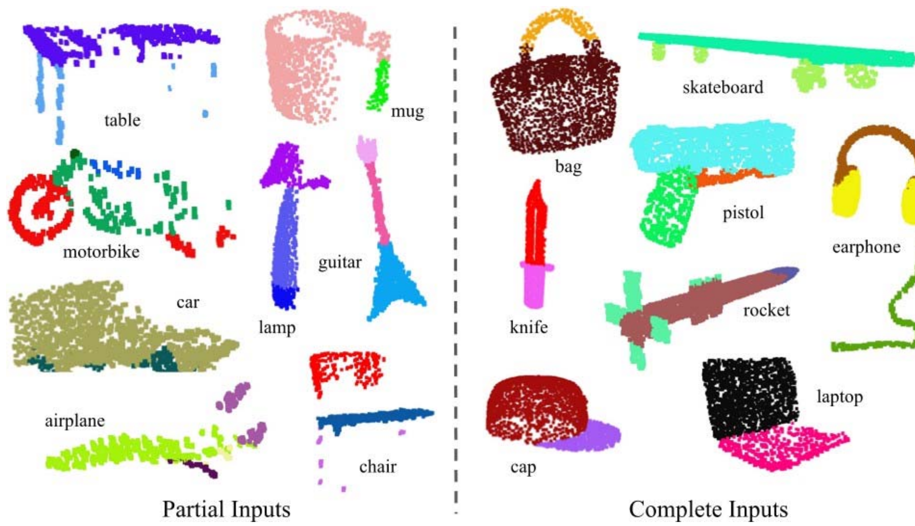
Results on Object Classification

	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 Object Part Segmentation

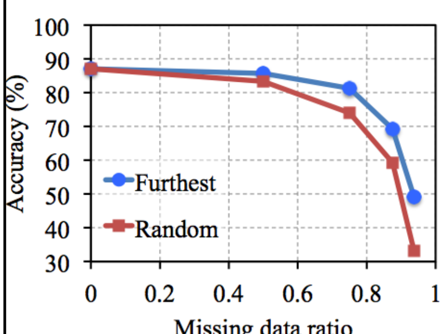
	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Wu [28]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [30]	81.4	81.0	78.4	77.7	**75.7**	87.6	61.9	**92.0**	85.4	**82.5**	**95.7**	**70.6**	91.9	**85.9**	53.1	69.8	75.3
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1
Ours	**83.7**	**83.4**	**78.7**	**82.5**	74.9	**89.6**	**73.0**	91.5	**85.9**	80.8	95.3	65.2	**93.0**	81.2	**57.9**	**72.8**	**80.6**

Dataset: ShapeNetPart; metric: mean IoU (%)

Results on Semantic Scene Parsing

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

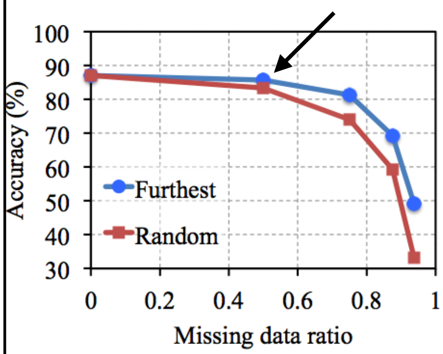
Robustness to Data Corruption



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

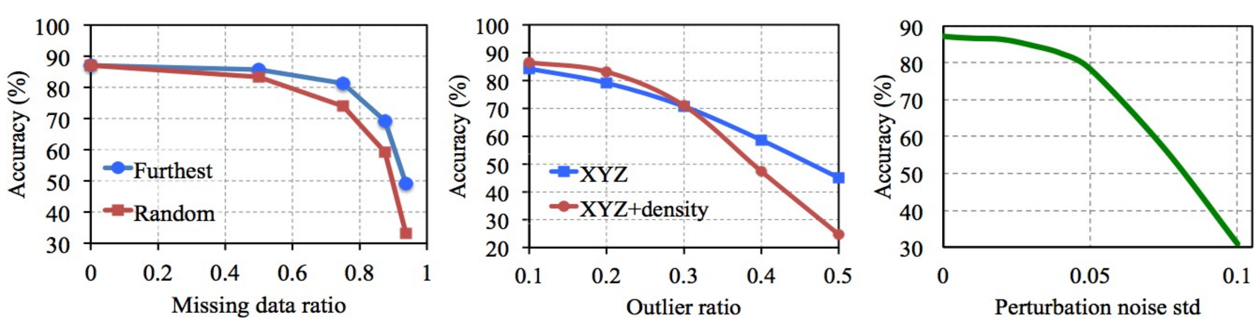
Robustness to Data Corruption

Less than 2% accuracy drop with 50% missing data



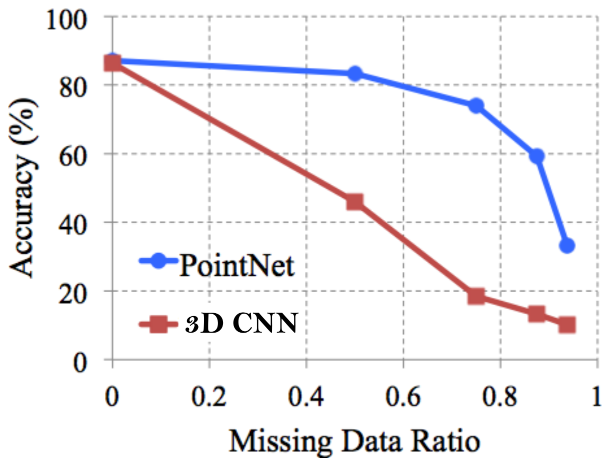
Dataset: ModelNet40; metric: 40-class classification accuracy (%)

Robustness to Data Corruption



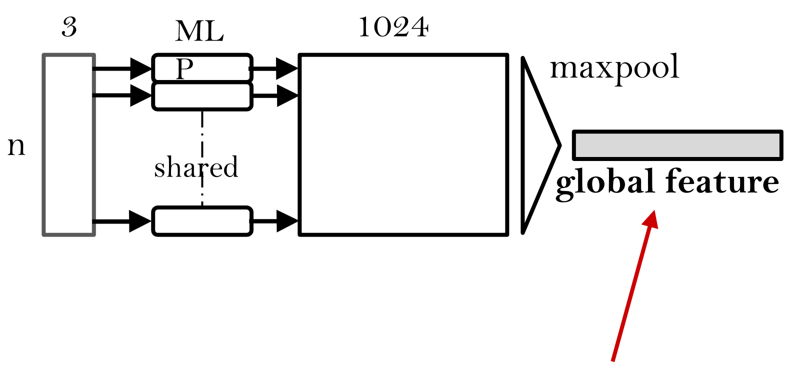
Dataset: ModelNet40; metric: 40-class classification accuracy (%)

Robustness to Data Corruption



Why is PointNet so robust to missing data?

Visualizing Global Point Cloud Features



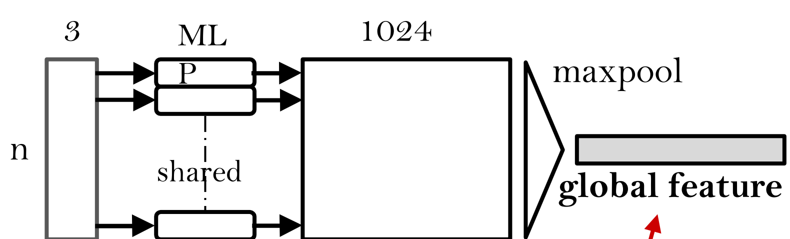
<u>Which input points</u> are contributing to the global feature?
(critical points)

Visualizing Global Point Cloud Features

Original Shape:

Critical Point Sets:

Visualizing Global Point Cloud Features



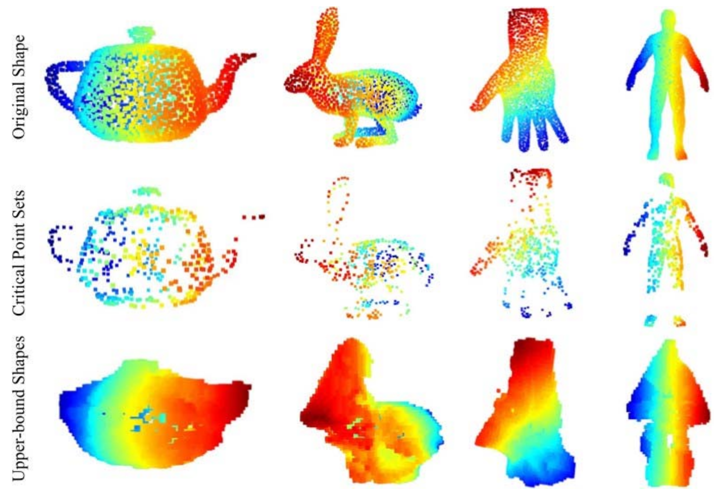
Which points won't affect the global feature?

Visualizing Global Point Cloud Features

Original Shape:

Critical Point Set:

Upper bound Set:

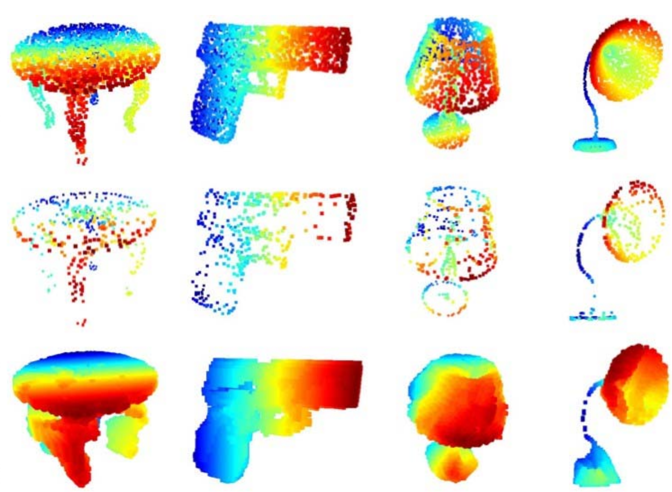


Visualizing Global Point Cloud Features

Original Shape:

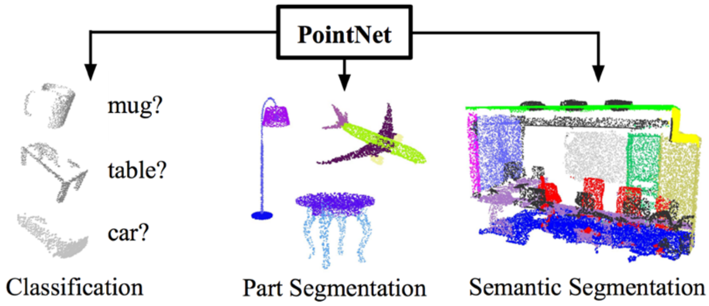
Critical Point Set:

Upper bound set:



Conclusion

- PointNet is a deep neural network that directly consumes point cloud.
- A unified approach to various 3D recognition tasks.



- Code & Data Avalable@*http://stanford.edu/~rqi/pointnet*

References

→ *R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "**PointNet**: Deep Learning on Point Sets for 3D Classification and Segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 77-85, DOI: 10.1109/CVPR.2017.16.*

→ ***Slide credit:** R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas (presentation at CVPR 2017)*

Thank You