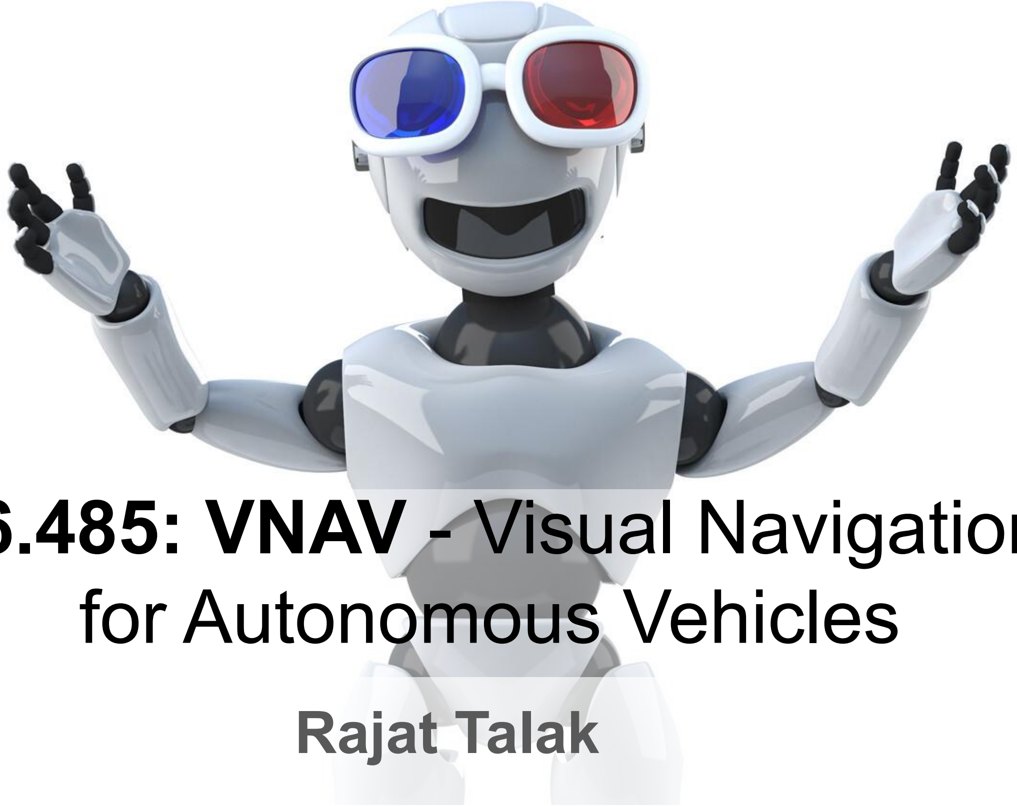


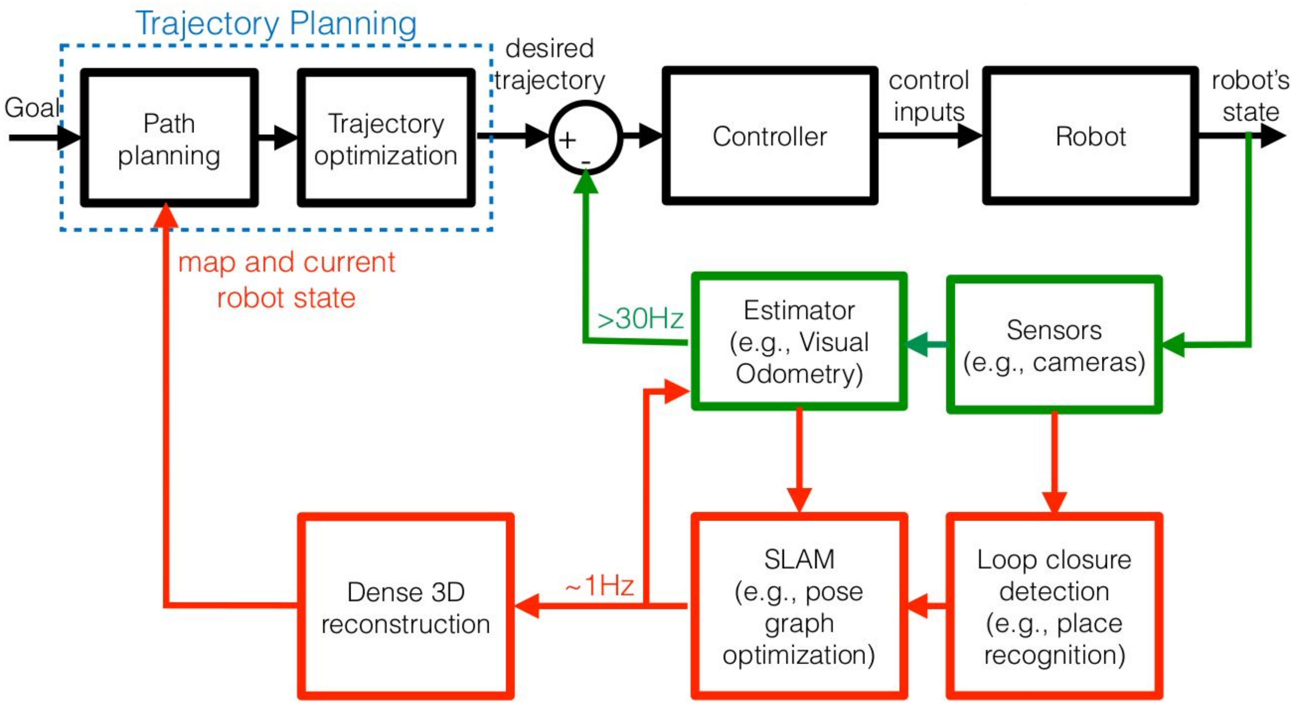
16.485: VNAV - Visual Navigation for Autonomous Vehicles

Rajat Talak

Lecture 31-32.5: Deep Learning Architectures on 3D Data

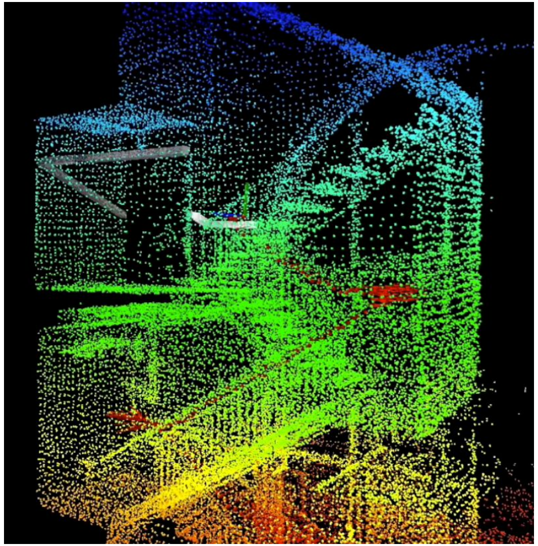
VNAV thus far …

3D Geometric Reconstruction

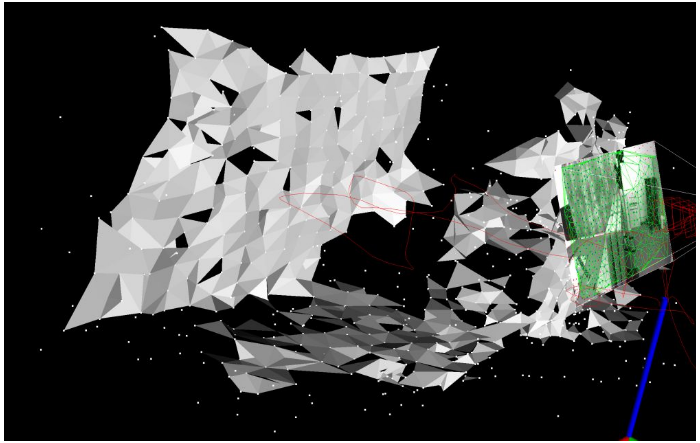


VNAV thus far …

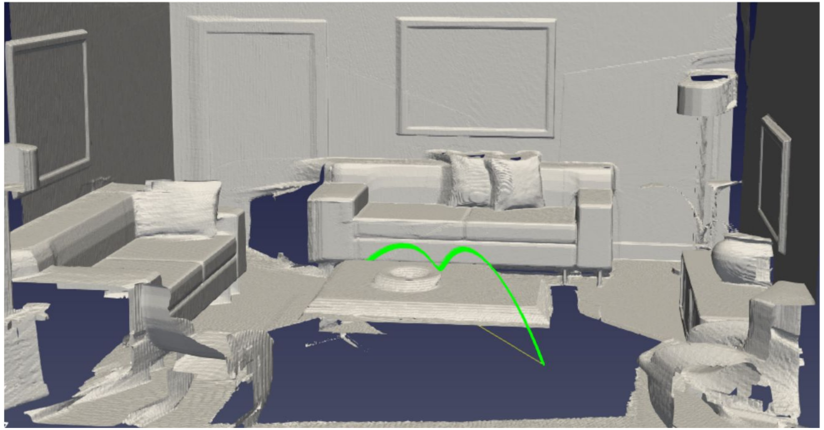
3D Geometric Reconstruction



Point Cloud



Mesh



Voxel

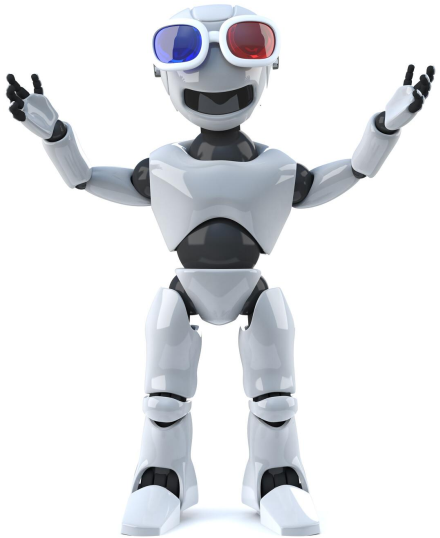
Vespa et al. "Efficient Octree-based Volumetric SLAM Supporting Signed-Distance and Occupancy Mapping" RAL 2017

Is this enough?

Is this enough? … No!!

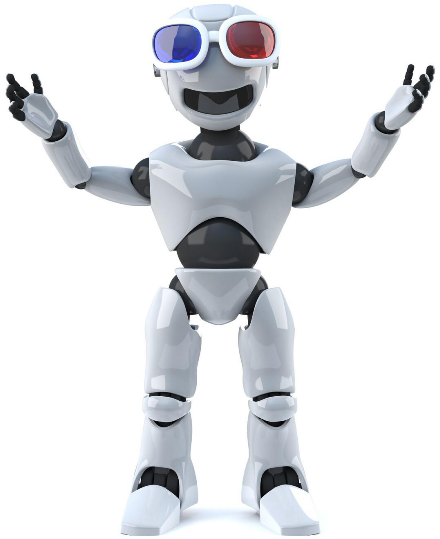
Is this enough? … No!!

“seeing is not understanding”



Is this enough? … No!!

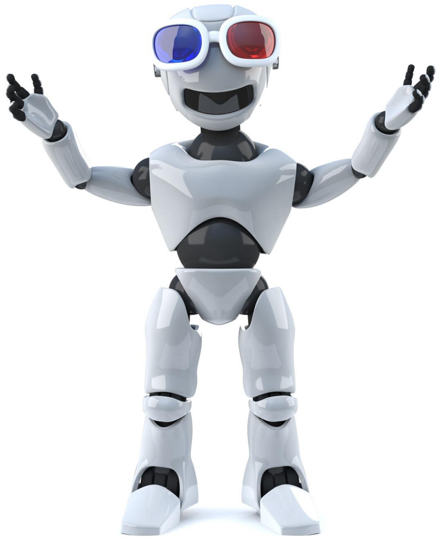
"seeing is not understanding"



Need to go beyond geometric reconstruction to 3D scene understanding

Is this enough?   … No!!

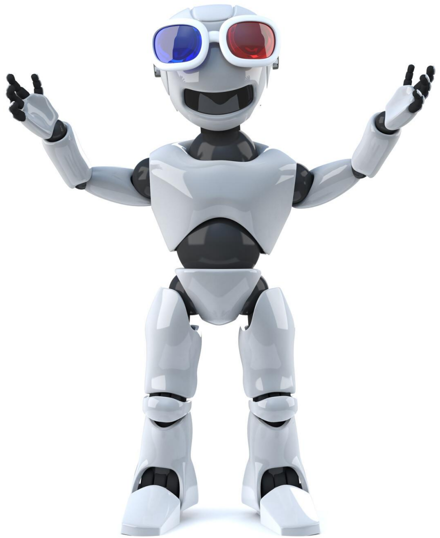
Detect objects and
humans



Need to go beyond geometric reconstruction to 3D scene understanding

Is this enough?   … No!!

Detect objects and humans



Learn interaction between human, object, scene

- Person reading a book
- Laptop is on the desk

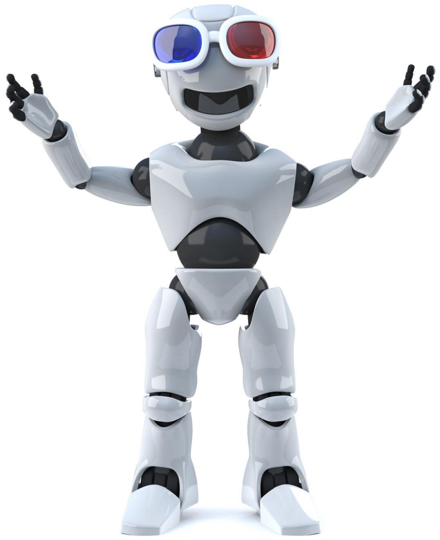
Need to go beyond geometric reconstruction to 3D scene understanding

Is this enough?   … No!!

Detect objects and humans

Interact with humans

- “Get me a cup of coffee”
- Surprise
- Human intent and emotions



Learn interaction between human, object, scene

- Person reading a book
- Laptop is on the desk

Need to go beyond geometric reconstruction to 3D scene understanding

Is this enough? … No!!

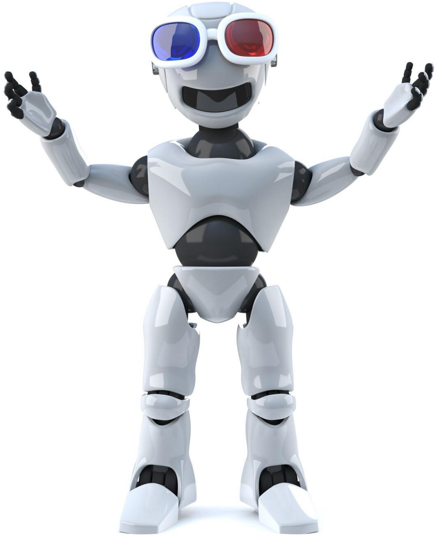
Detect objects and humans

Interact with humans

- "Get me a cup of coffee"
- Surprise
- Human intent and emotions

Scene dynamics

- Falling object



Learn interaction between human, object, scene

- Person reading a book
- Laptop is on the desk

Need to go beyond geometric reconstruction to 3D scene understanding

Is this enough?   … No!!

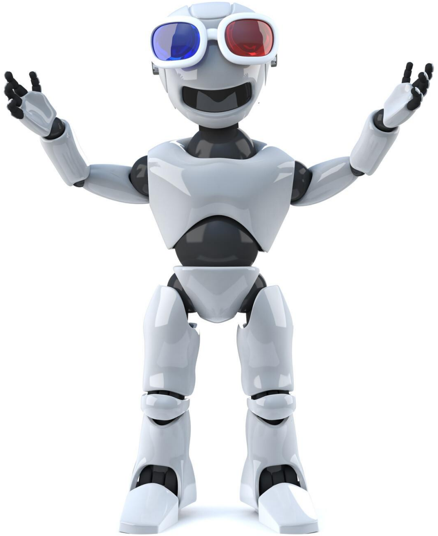
Detect objects and humans

Interact with humans

- "Get me a cup of coffee"
- Surprise
- Human intent and emotions

Scene dynamics

- Falling object



Learn interaction between human, object, scene

- Person reading a book
- Laptop is on the desk

Identify and work with deformable objects

- Distinguish table cloth from the table
- "Spread the table cloth on the table"f

Need to go beyond geometric reconstruction to 3D scene understanding

Is this enough? … No!!

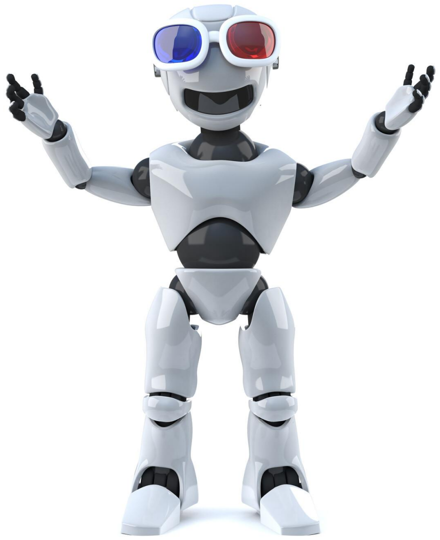
Detect objects and humans

Interact with humans

- “Get me a cup of coffee”
- Surprise
- Human intent and emotions

Scene dynamics

- Falling object



Learn interaction between human, object, scene

- Person reading a book
- Laptop is on the desk

Identify and work with deformable objects

- Distinguish table cloth from the table
- “Spread the table cloth on the table”f

Need for Semantic Understanding of the 3D Scene

Semantic Understanding

No formal definition

Semantic Understanding

No formal definition

"... we consider semantics in a robotics context to be about the meaning of things; the meaning of places, objects, other entities occupying the environment, or even language used in communicating between robots and humans or between robots themselves."

Garg et al. "Semantics for Robotic Mapping, Perception and Interaction: A Survey" 2021

Semantic Understanding

No formal definition

"... we consider semantics in a robotics context to be about the meaning of things; the meaning of places, objects, other entities occupying the environment, or even language used in communicating between robots and humans or between robots themselves."

Garg et al. "Semantics for Robotic Mapping, Perception and Interaction: A Survey" 2021

"... the research focus has shifted from reconstructing the 3D scene geometry to enhancing the 3D maps with semantic information about scene components."

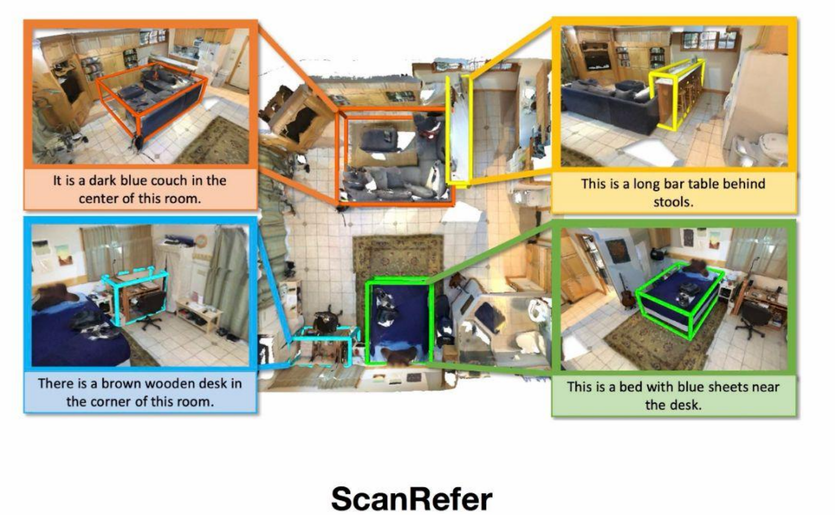
Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari
"SceneGraphFusion: Incremental 3D Scene Graph Prediction from RGB-D Sequences" 2021

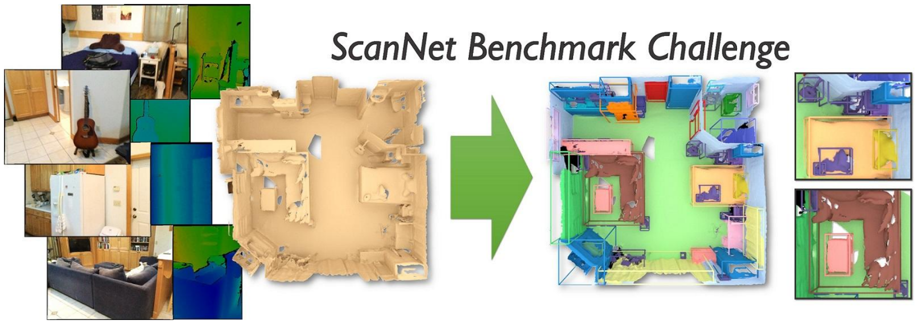
Research Activity

1st Workshop on Language for 3D Scenes at CVPR 2021

3D Scene Understanding for Vision, Graphics, and Robotics at CVPR 2021

3rd ScanNet Indoor Scene Understanding Challenge at CVPR 2021

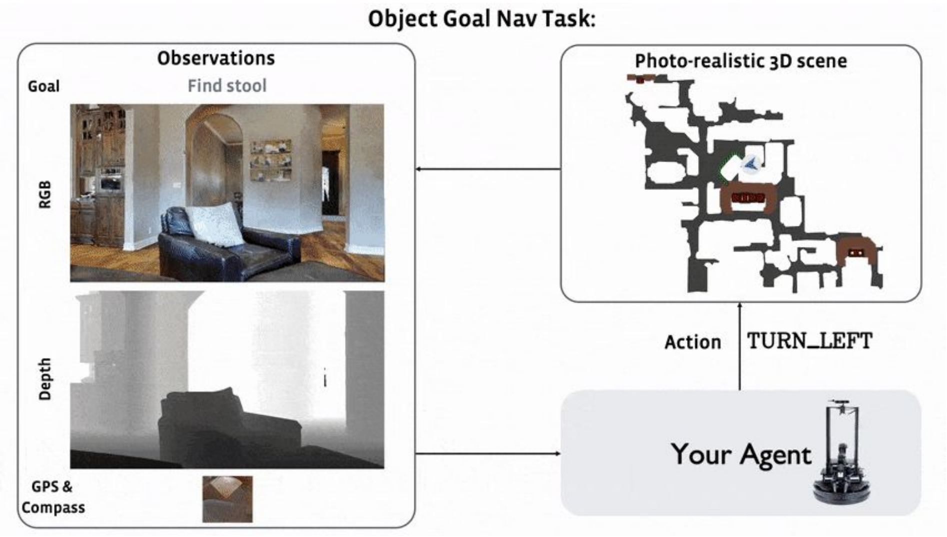




Research Activity

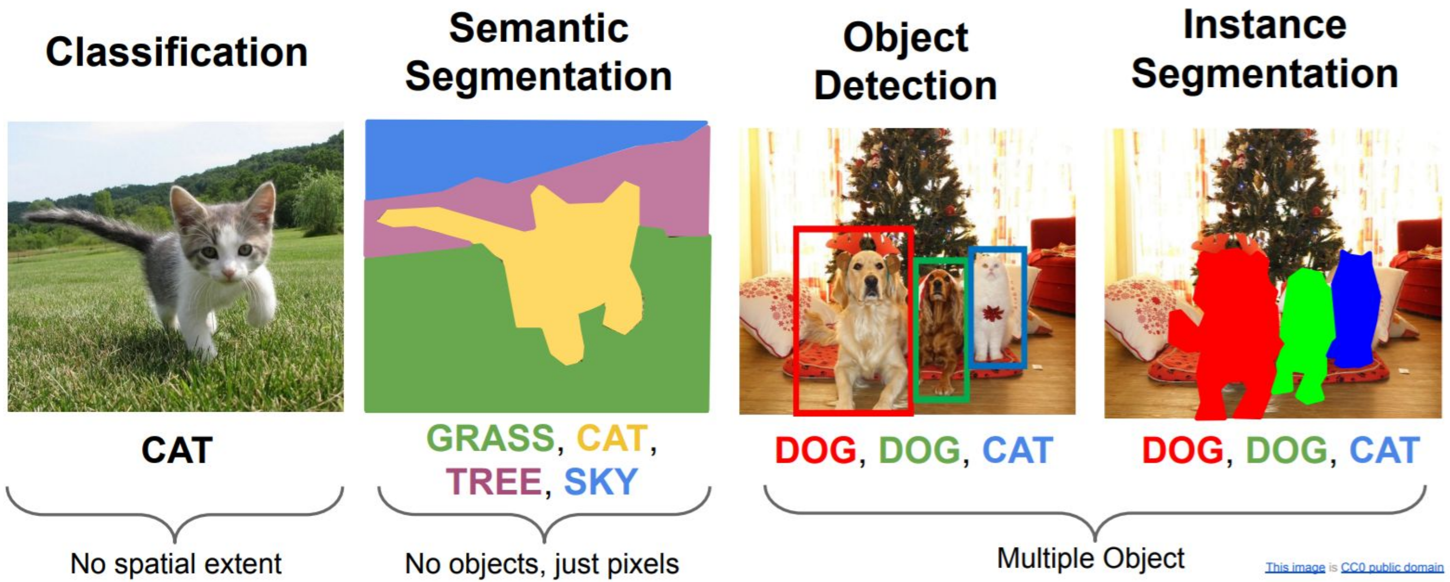
Facebook AI Habitat Challenge

Given an object, the goal is to move and find an instance of it in the scene.



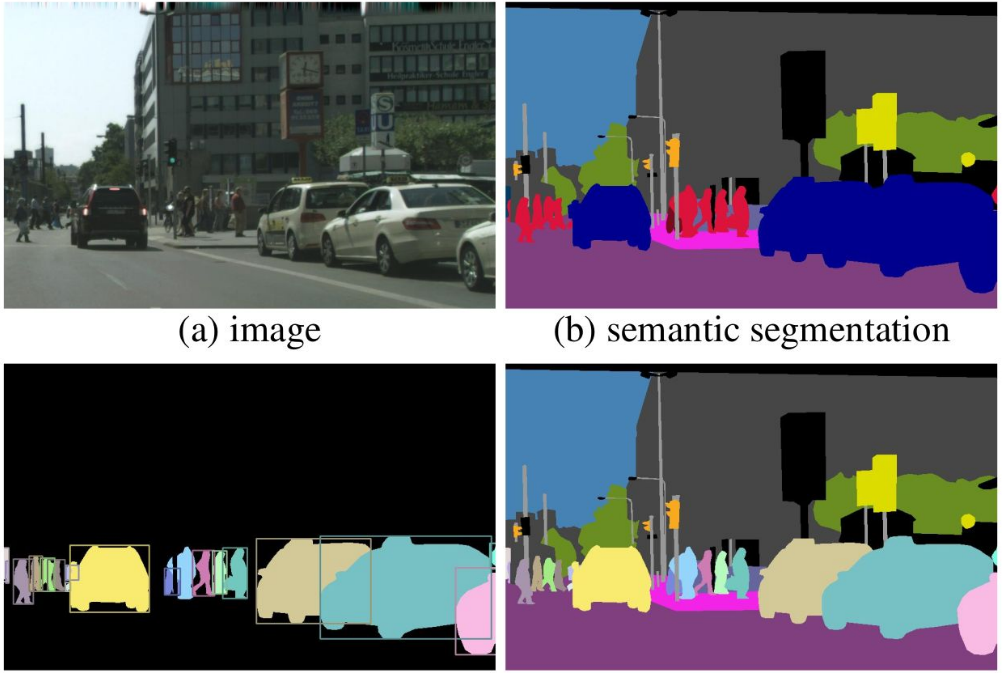
source : https://aihabitat.org/challenge/2021/

Semantic Understanding in Images



Source: Fei-Fei Li, Rajat Krishna, Danfei Xu "Stanford CS231n: Convolutional Neural Networks for Visual Recognition" Spring 2021

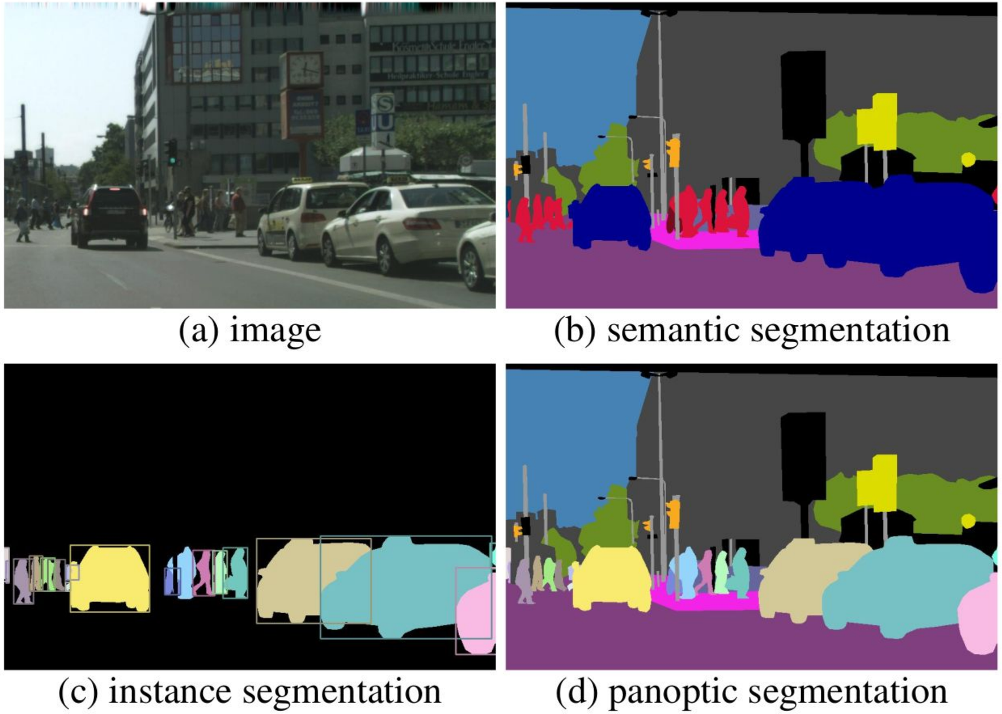
Semantic Understanding in Images



(a) image (b) semantic segmentation (c) instance segmentation (d) panoptic segmentation

Source: Kirillov et al. "Panoptic Segmentation" 2019

Semantic Understanding in Images

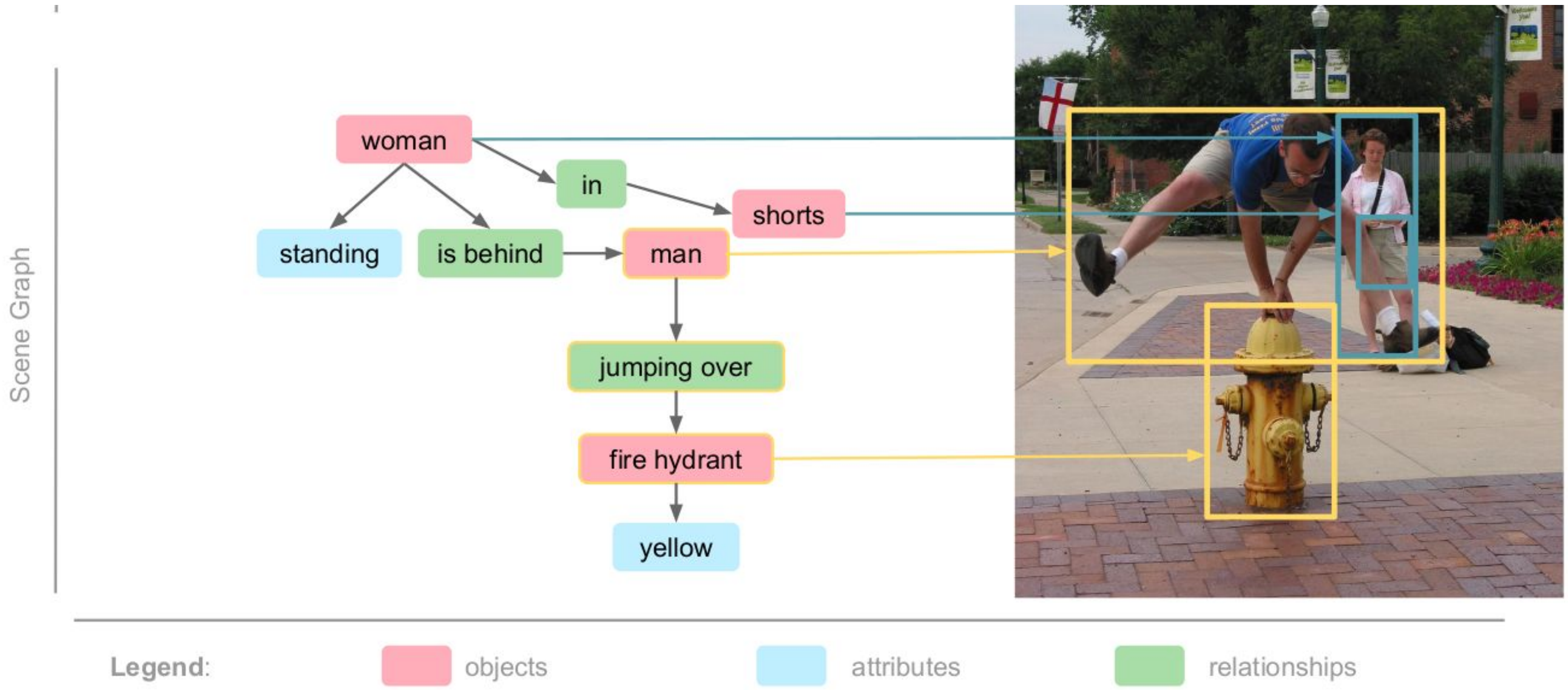


Source: Kirillov et al. “Panoptic Segmentation” 2019

Recently, panoptic segmentation approaches have been used in volumetric mapping pipelines.

Schmid et al. “Panoptic Multi-TSDFs: a Flexible Representation for Online Multi-resolution Volumetric Mapping and Long-term Dynamic Scene Consistency” 2021

Semantic Understanding in Images



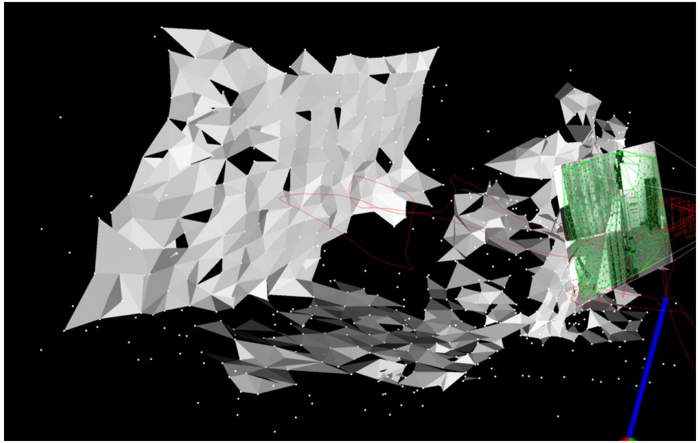
Source: Rajay Krishna et al. “Visual Genome: Connecting Language and Vision using Crowdsourced Dense Image Annotations” 2016

Semantic Understanding in Images

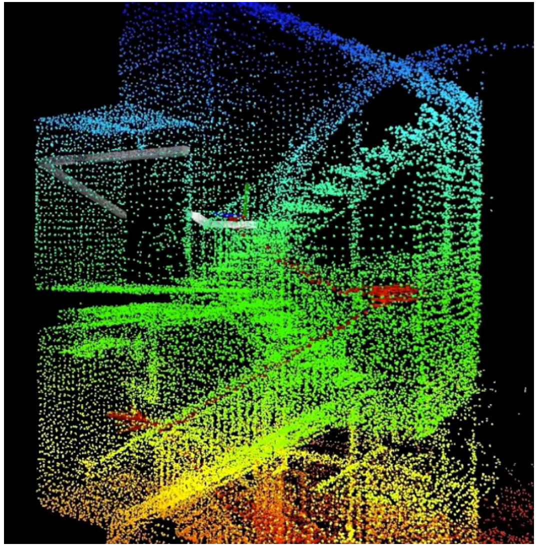
State-of-the-art approaches use
Deep Learning based architectures

Semantic Understanding on 3D Data

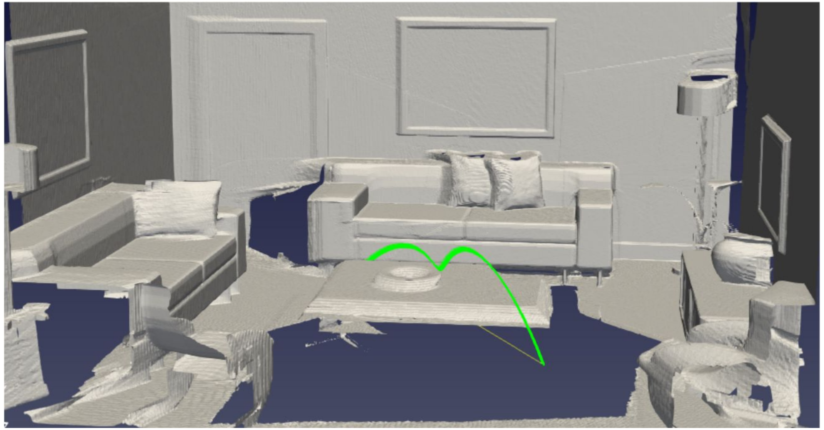
Point Clouds, Voxels, Meshes



Mesh



Point Cloud

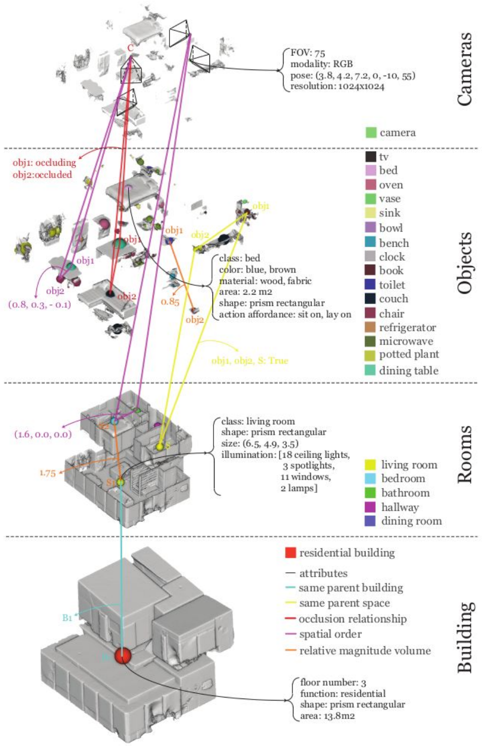


Voxel

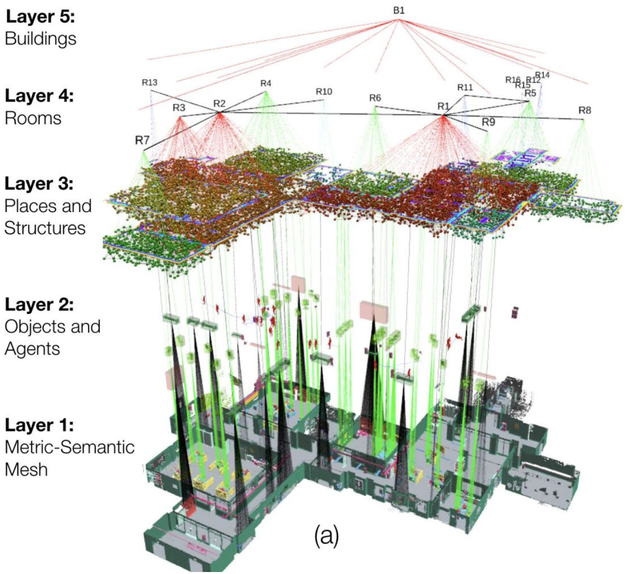
Vespa et al. "Efficient Octree-based Volumetric SLAM Supporting Signed-Distance and Occupancy Mapping" RAL 2017

Semantic Understanding on 3D Data

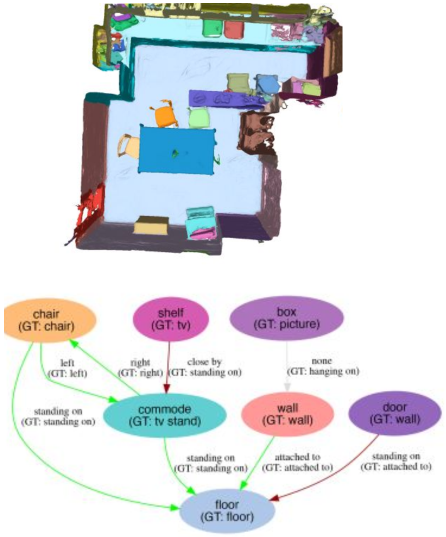
Graphs



Armeni et al. “3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera” 2019



Rosinol et al. “3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans” 2020



Wald et al. “Learning 3D Semantic Scene Graphs from 3D Indoor Reconstruction” 2020

Semantic Understanding on 3D Data

How do we develop Deep Learning Architectures on Voxels, Point Clouds, Meshes, and Graphs?

Plan for the three lectures …

1 Deep Learning Architectures on 3D Data

- Motivation: Semantic Understanding
- Recap: Machine Learning, Deep Learning on Image
- Neural Architectures for 3D Data
    - Voxels, Point clouds, Meshes
- Datasets and Software

2 Geometric Deep Learning

- Unifying view of developing architectures on all data
- Symmetry
- Equivariance, Invariance, Convolutions
- Unified Blueprint

3 Learning on Scene Graphs

- Scene Graphs for Semantic Understanding
- Graph Neural Networks
- Limitations
- Node and Relationship Prediction

First Part

1

Deep Learning Architectures on 3D Data

- Motivation: Semantic Understanding
- Recap: Machine Learning, Deep Learning on Image
- Neural Architectures for 3D Data
  - Voxels, Point clouds, Meshes
- Datasets and Software

Key ideas and heuristics for Deep Learning architectures on Voxels, Point Clouds, Meshes

First Part

1

Deep Learning Architectures on 3D Data

- Motivation: Semantic Understanding
- Recap: Machine Learning, Deep Learning on Image
- Neural Architectures for 3D Data
  - Voxels, Point clouds, Meshes
- Datasets and Software

Key ideas and heuristics for Deep Learning architectures on Voxels, Point Clouds, Meshes

Background ...

A Quick Recap:
The Machine Learning Problem

The Machine Learning Problem

Data $\{(x_i, y_i)\}_{i=1}^{N}$ $\quad x_i \in \mathbb{X} \quad y_i \in \mathbb{Y}$

Truth $f^* : \mathbb{X} \to \mathbb{Y}$

Model $f_\theta : \mathbb{X} \to \mathbb{Y} \qquad \theta \in \Theta$

Goal: find $\theta \in \Theta$ such that $f^* \approx f_\theta$

The Machine Learning Problem

Loss Function $l : \mathbb{Y} \times \mathbb{Y} \rightarrow \mathbb{R}$

$$l(y, y') = ||y - y'||_2^2$$

$$l(y, y') = -y \log(y')$$

Empirical Loss Minimization

$$\min_{\theta \in \Theta} \mathcal{L}_\theta = \frac{1}{N} \sum_{i=1}^{N} l(y_i, f_\theta(x_i))$$

Optimization Method

Gradient descent $\theta_{t+1} = \theta_t - \alpha_t \partial \mathcal{L}_\theta / \partial \theta$

learning rate

The Goal

Come up with a model $f_\theta : \mathbb{X} \rightarrow \mathbb{Y}$ such that $f^* \approx f_\theta$

Terminology

Come up with a model $f_\theta : \mathbb{X} \to \mathbb{Y}$ such that $f^* \approx f_\theta$

Architecture

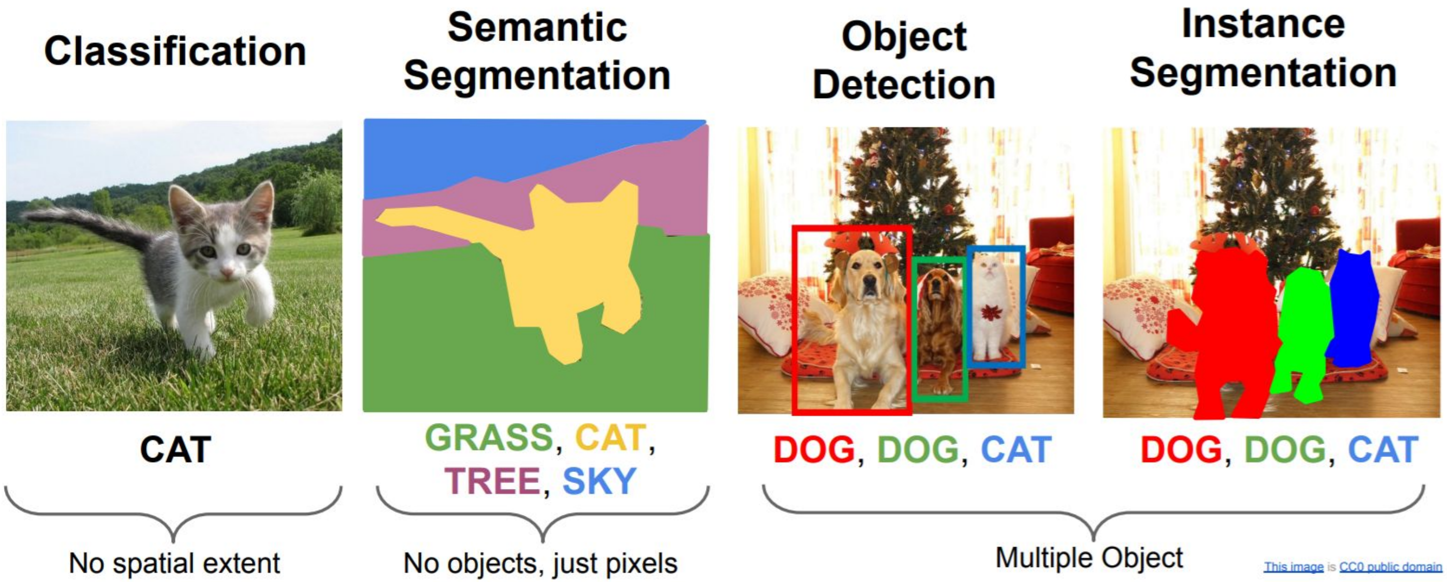
$$\mathcal{A} = \{f_\theta : \mathbb{X} \to \mathbb{Y} \mid \theta \in \Theta\}$$

Model

f_θ for a particular choice of θ

A Quick Recap:
Deep Learning Architectures on Images

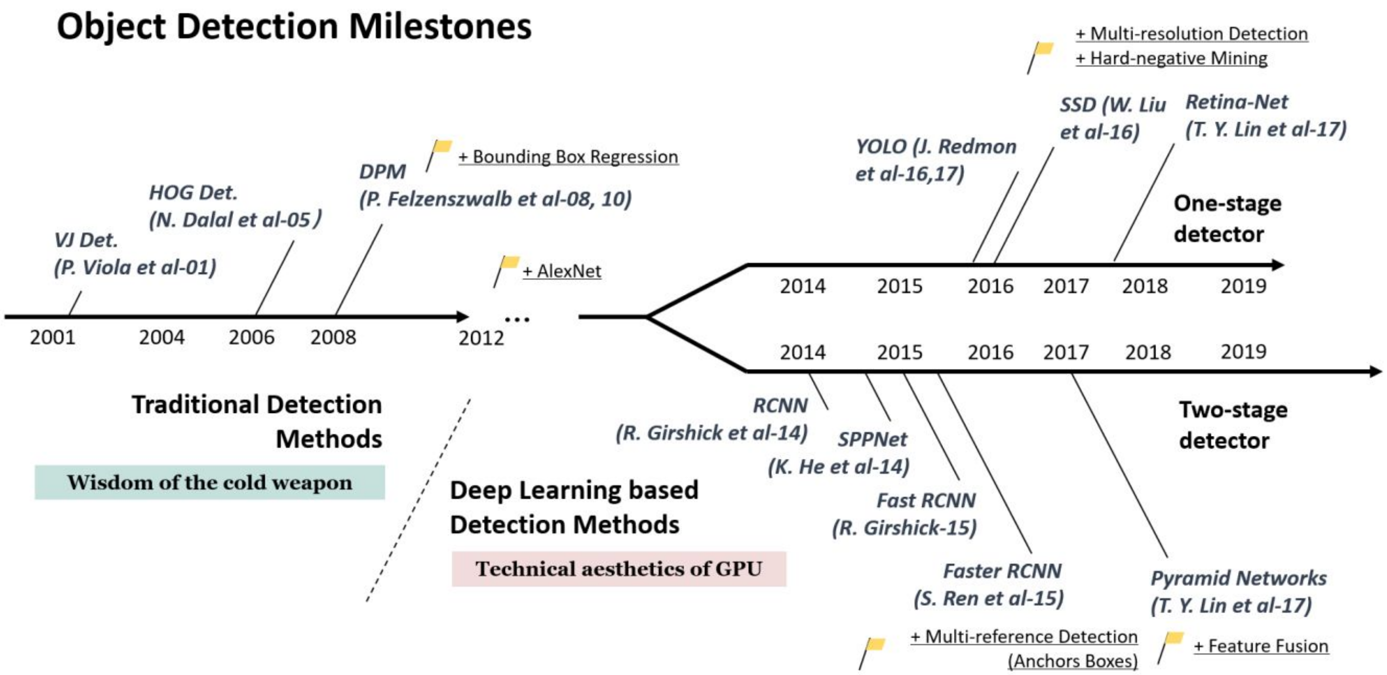
Semantic Understanding on Images

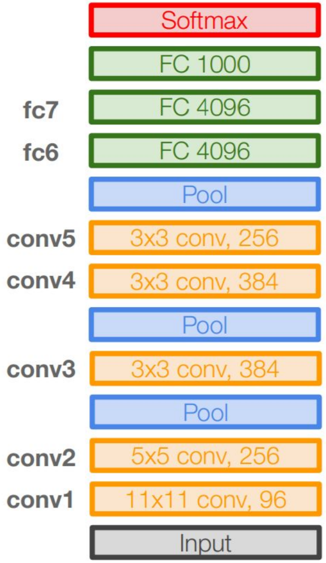


$\mathbb{X}, \mathbb{Y}?$

Source: Fei-Fei Li, Rajat Krishna, Danfei Xu "Stanford CS231n: Convolutional Neural Networks for Visual Recognition" Spring 2021

Progress on Object Detection (20 years)



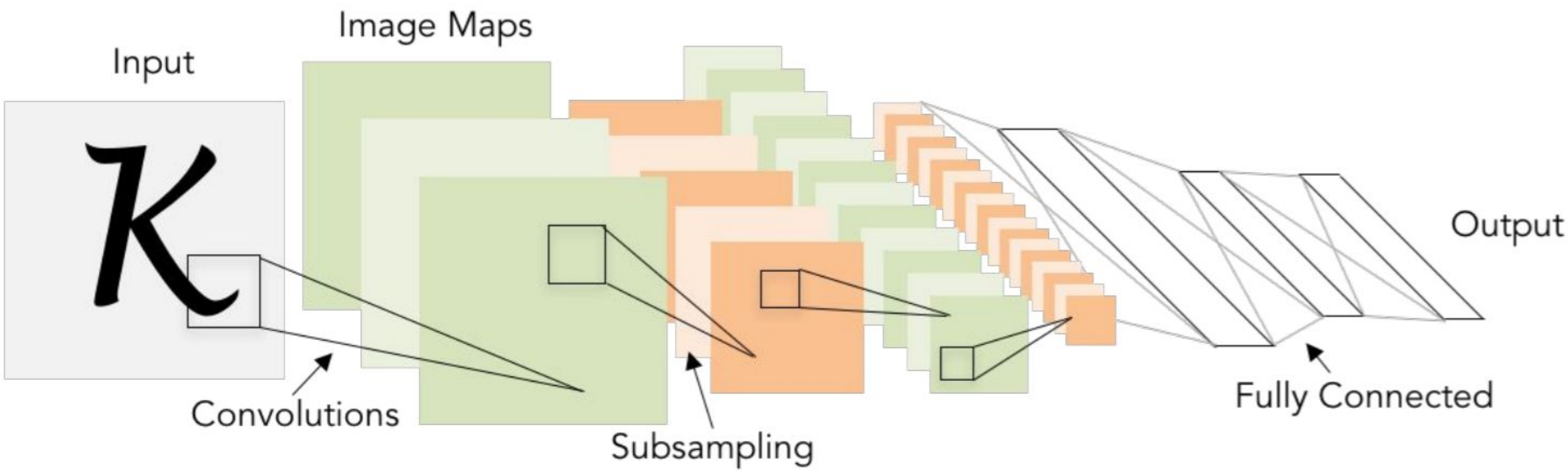


AlexNet

State-of-the-art models = composition of convolution, pooling, unpooling, fully connected layers

Zou et al. "Object Detection in 20 Years: A Survey" 2019

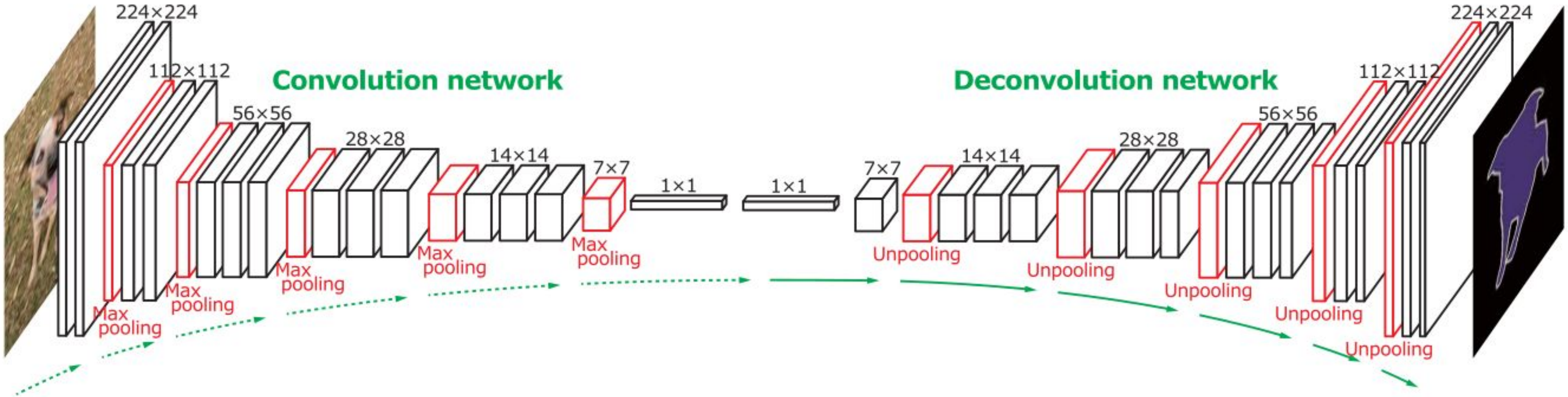
Convolutional Neural Networks for Classification



LeCun et al “Gradient-based Learning Applied to Document Recognition” 1998

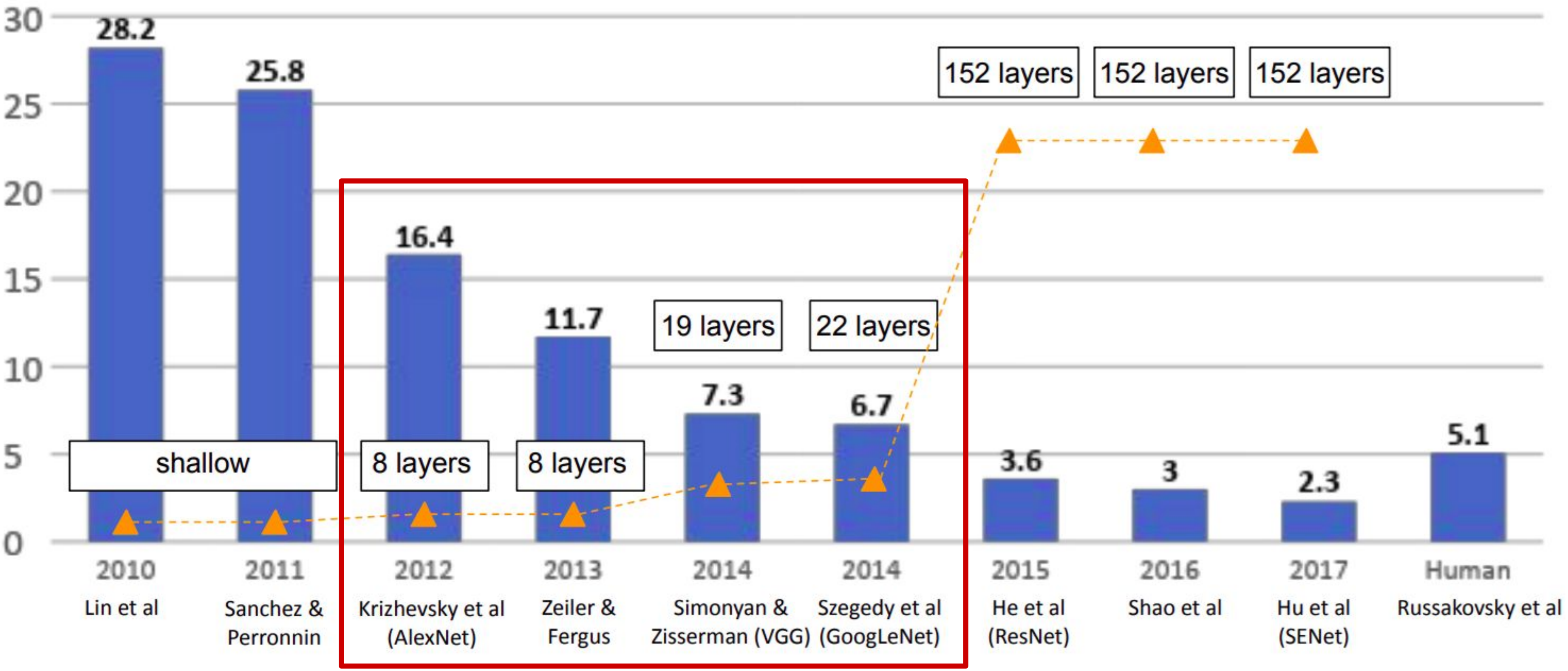
State-of-the-art models = composition of convolution, pooling, unpooling, fully connected layers

Convolutional Neural Networks for Segmentation



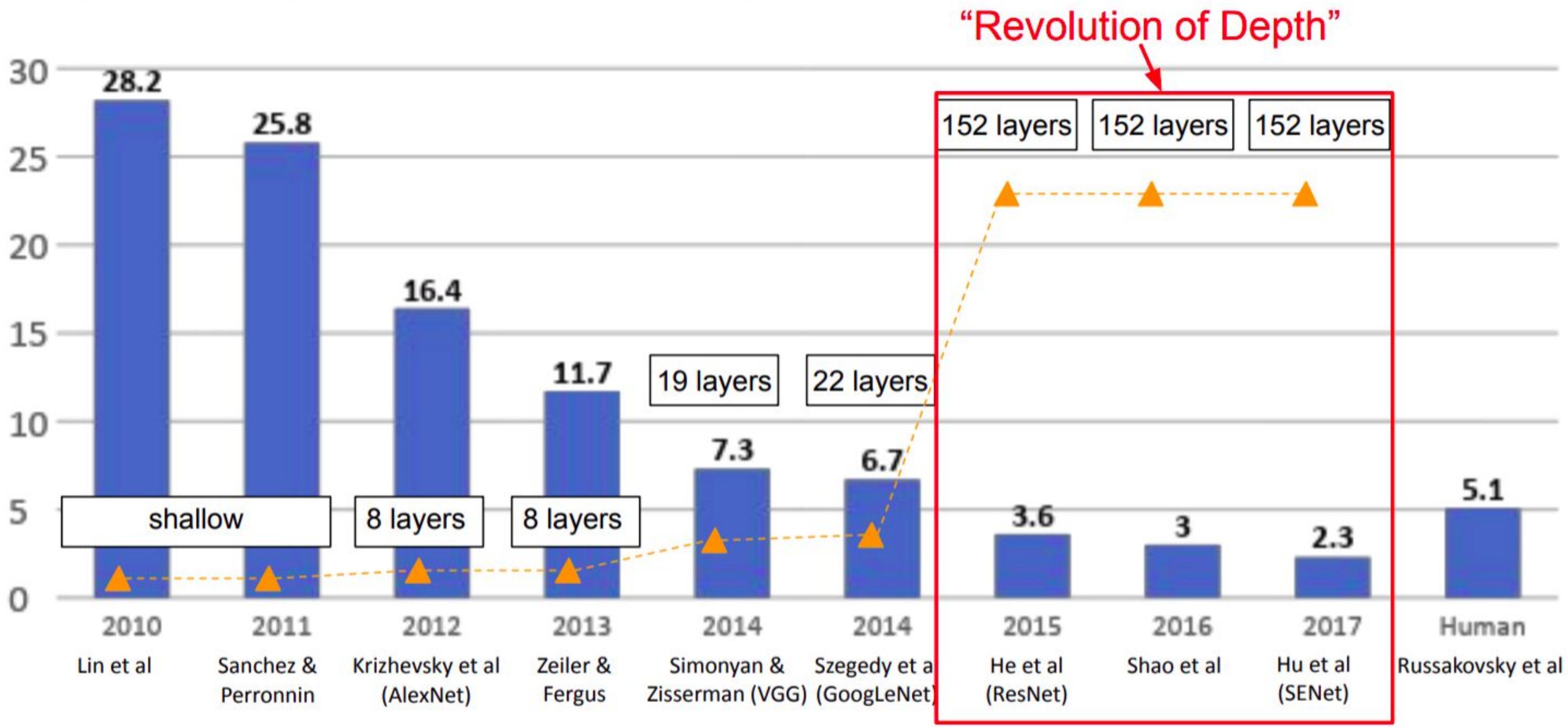
Noh et al “Learning Deconvolution Network for Semantic Segmentation” CVPR 2015

ImageNet Large Scale Visual Recognition Challenge



Source: Fei-Fei Li, Rajat Krishna, Danfei Xu “Stanford CS231n: Convolutional Neural Networks for Visual Recognition” Spring 2021

ImageNet Large Scale Visual Recognition Challenge



Source: Fei-Fei Li, Rajat Krishna, Danfei Xu "Stanford CS231n: Convolutional Neural Networks for Visual Recognition" Spring 2021

Residual Connections

- Deeper models were harder to optimize

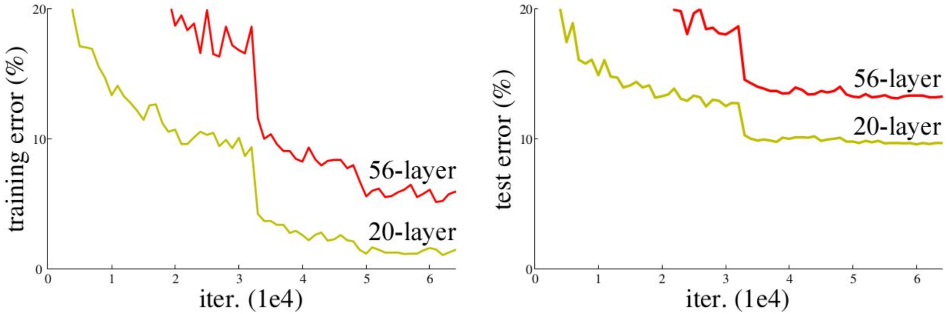
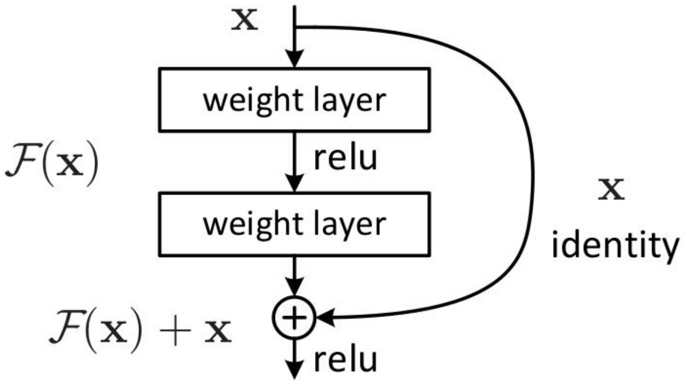


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error.

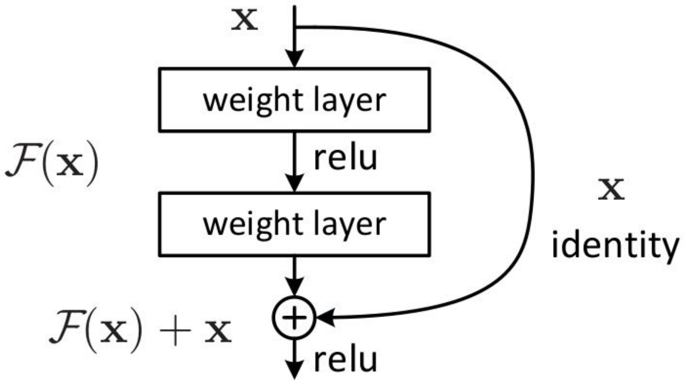
He et al. “Deep Residual Learning for Image Recognition” 2015

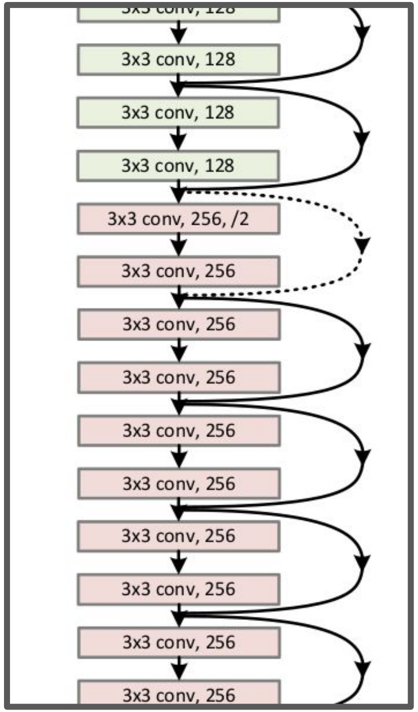
Residual Connections



He et al. “Deep Residual Learning for Image Recognition” 2015

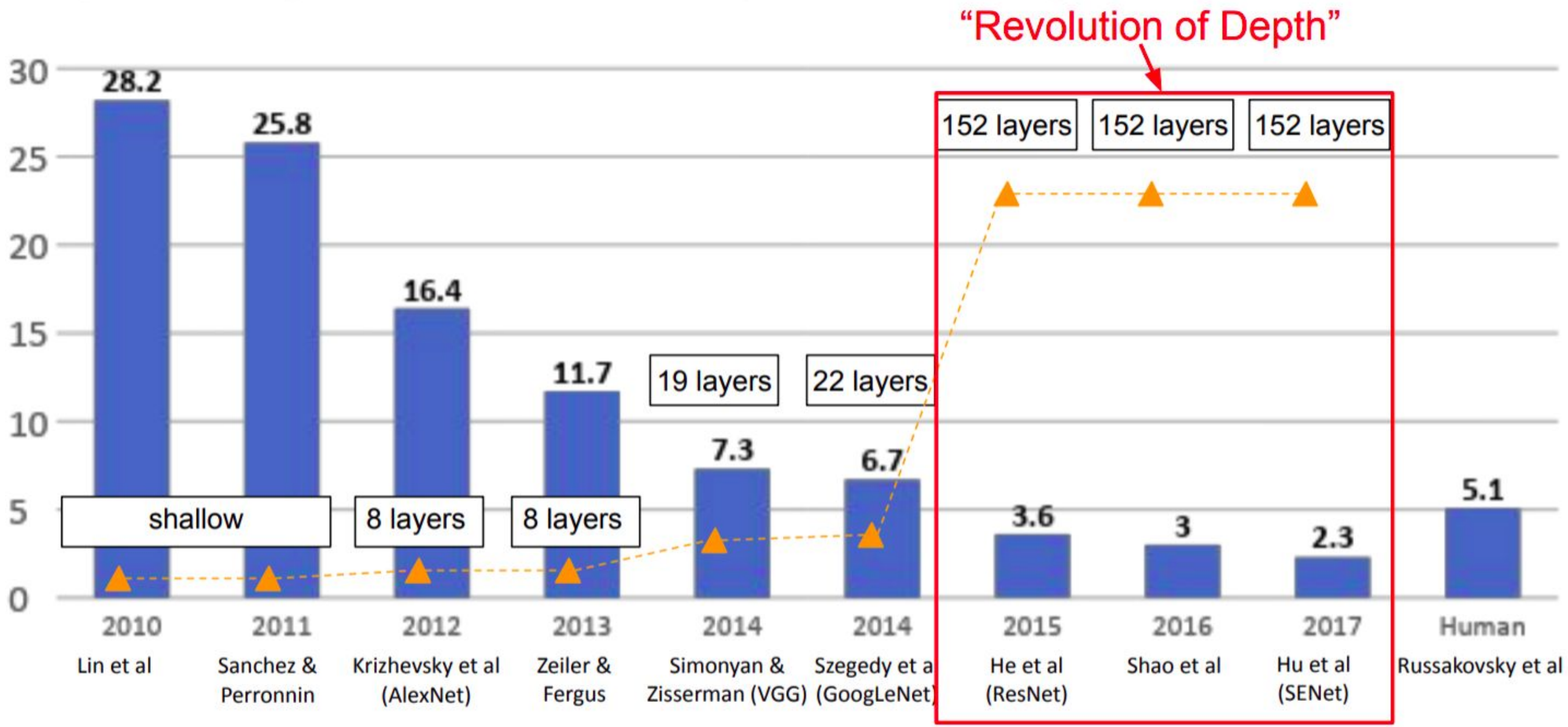
Residual Connections





He et al. “Deep Residual Learning for Image Recognition” 2015

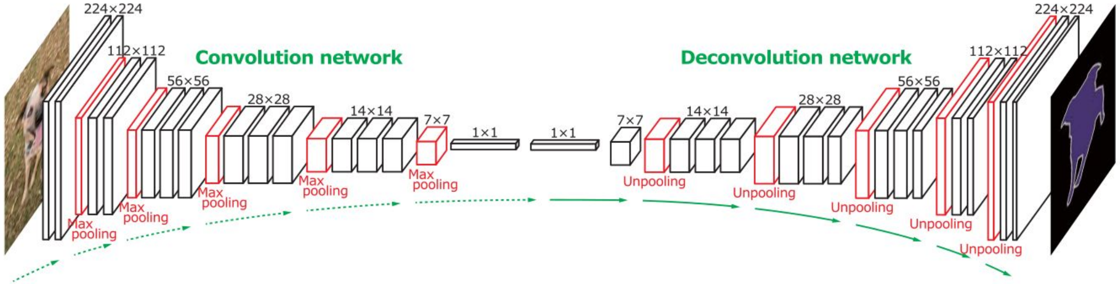
ImageNet Large Scale Visual Recognition Challenge



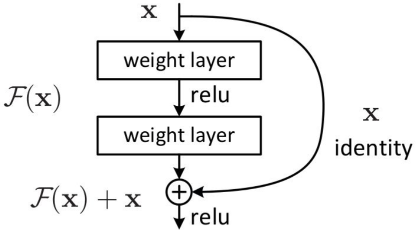
Source: Fei-Fei Li, Rajat Krishna, Danfei Xu "Stanford CS231n: Convolutional Neural Networks for Visual Recognition" Spring 2021

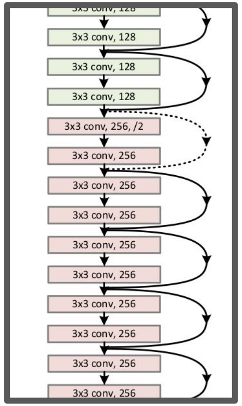
Takeaways ...

- Basic building blocks:
    - Convolutions
    - Pooling
    - Unpooling
    - Single and Multi-layer perceptron
- Residual Connections



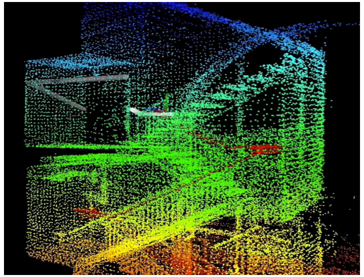
Noh et al “Learning Deconvolution Network for Semantic Segmentation” CVPR 2015



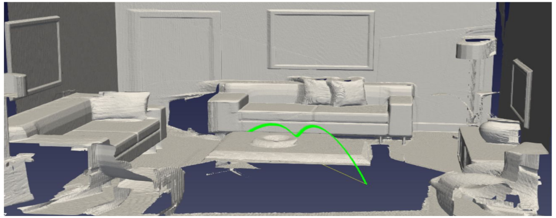


He et al. “Deep Residual Learning for Image Recognition” 2015

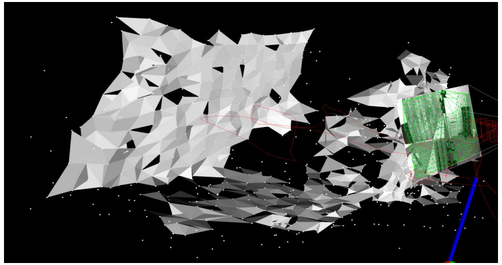
Architectures for Learning in 3D



Point Cloud



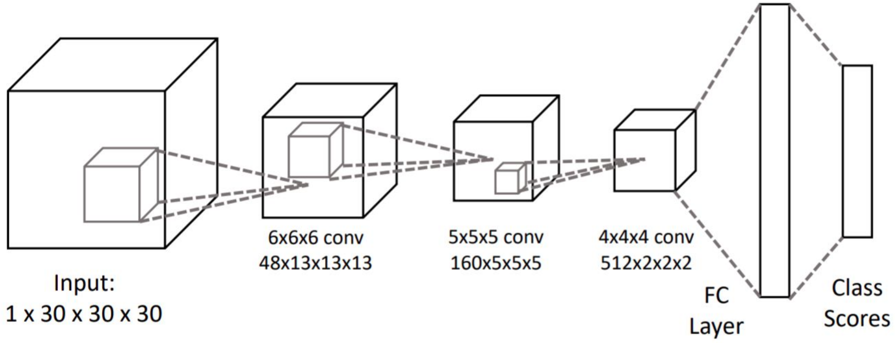
Voxel



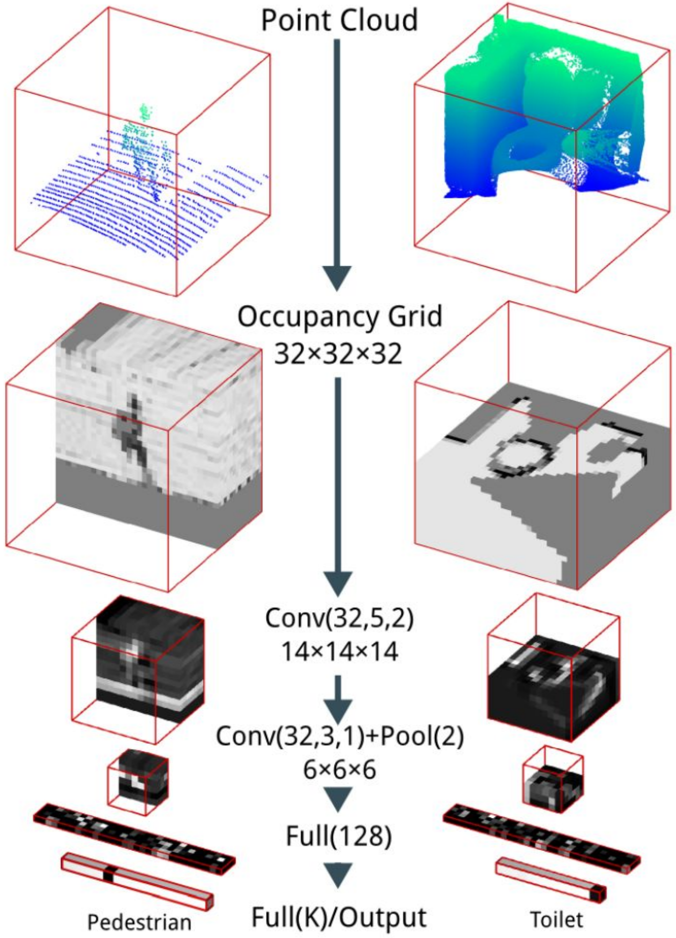
Mesh

Voxels

Convolutions on Voxel Grids



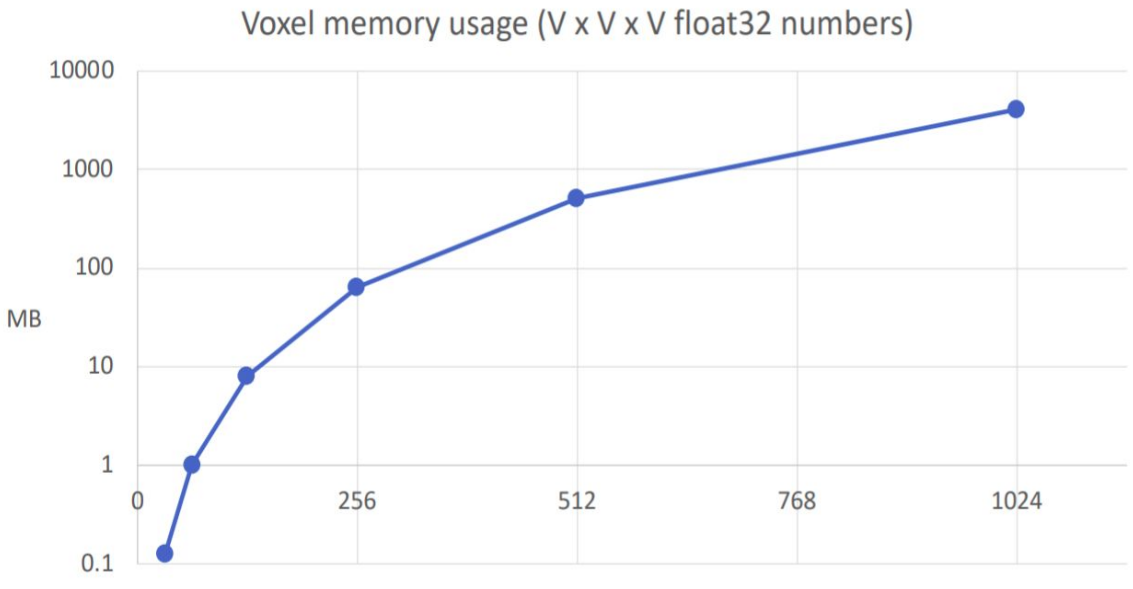
Wu et al "3D ShapeNets: A Deep Representation for Volumetric Shapes" CVPR 2015



Naturana and Scherer "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition" IROS 2015

Limitations

Very high memory usage



Source: Justin Johnson "Deep Learning for Computer Vision" Michigan University, Fall 2020.

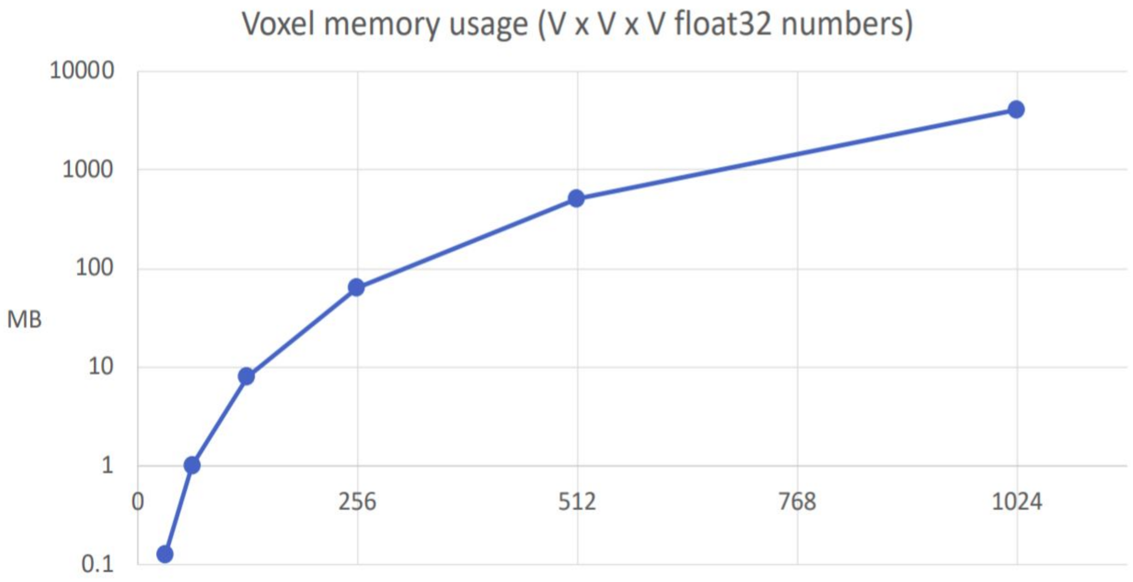
Storing 1024^3 voxel grid takes 4GB of memory

Limitations

Very high memory usage

Reported results on small sized voxel grids 32^3

Naturana and Scherer "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition" IROS 2015



Source: Justin Johnson "Deep Learning for Computer Vision" Michigan University, Fall 2020.

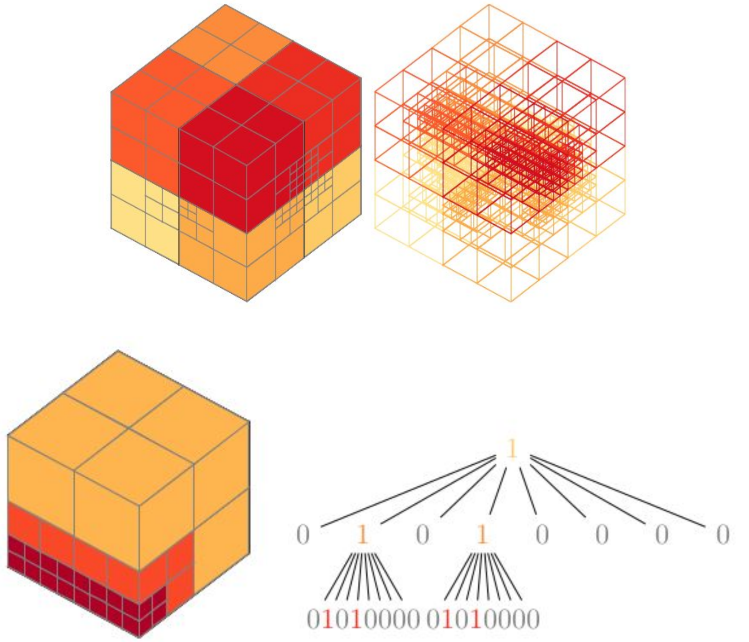
Storing 1024^3 voxel grid takes 4GB of memory

Octree-based Architectures

Define convolutions on octree

Helps due to sparsity of occupied regions. But not much!

Reported results on voxel grids of size 256^3

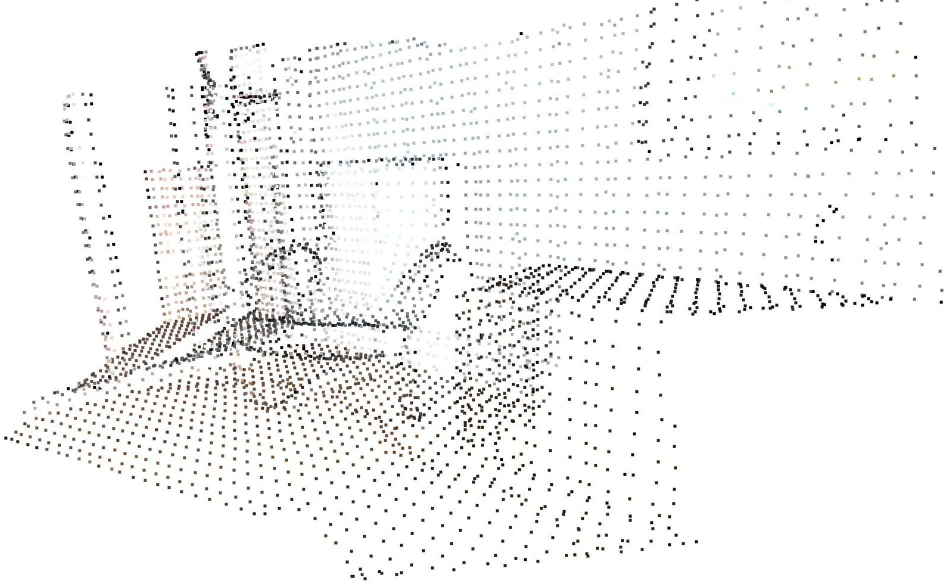


Source: Riegler et al. “OctNet: Learning Deep 3D Representations at High Resolution” 2017

Point Clouds

Have more inherent structure than voxel representation

Representative of the sparse data

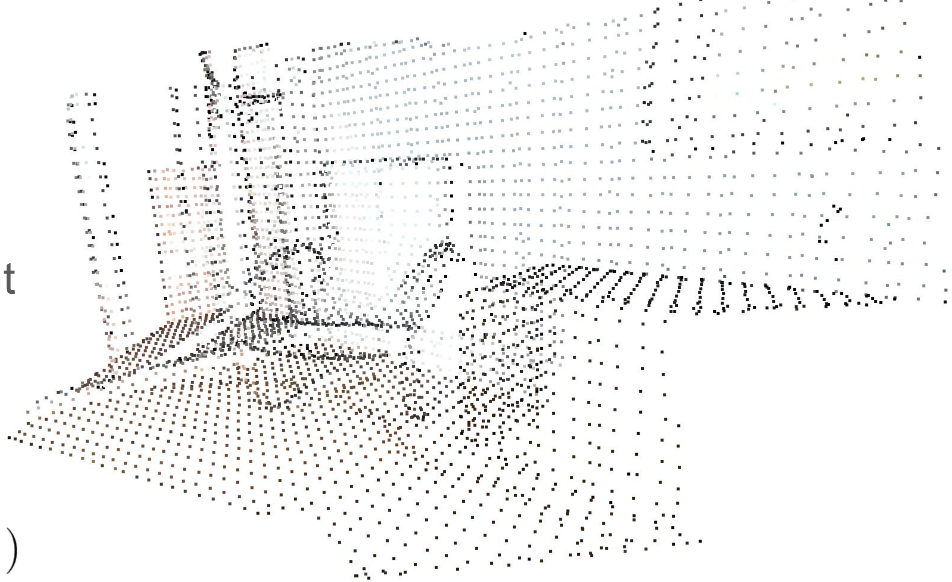


Point Clouds

PointNet

PointNet

The (classification) output should be invariant to ordering of points in the point cloud.

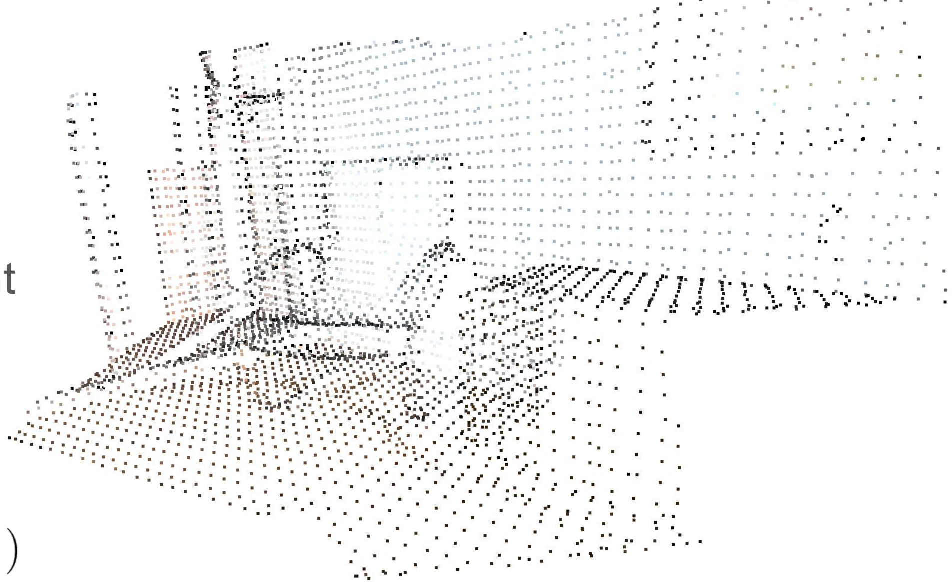


$$f(\{x_1, x_2, \ldots x_n\}) = g(h(x_1), h(x_2), \ldots h(x_n))$$

Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” CVPR 2017

PointNet

The (classification) output should be invariant to ordering of points in the point cloud.

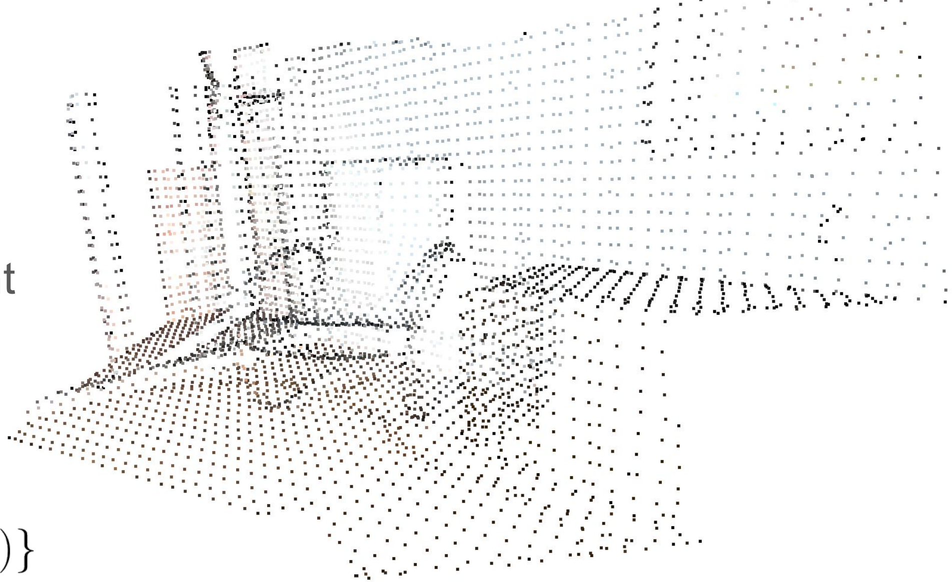


$$f(\{x_1, x_2, \ldots x_n\}) = g(h(x_1), h(x_2), \ldots h(x_n))$$

max pooling

Qi et al. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" CVPR 2017

PointNet



The (classification) output should be invariant to ordering of points in the point cloud.

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

Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” CVPR 2017

PointNet: Basic Operations

MLP + Max Pooling

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

Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” CVPR 2017

PointNet: Basic Operations

MLP + Max Pooling

shared weights

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

Qi et al. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" CVPR 2017

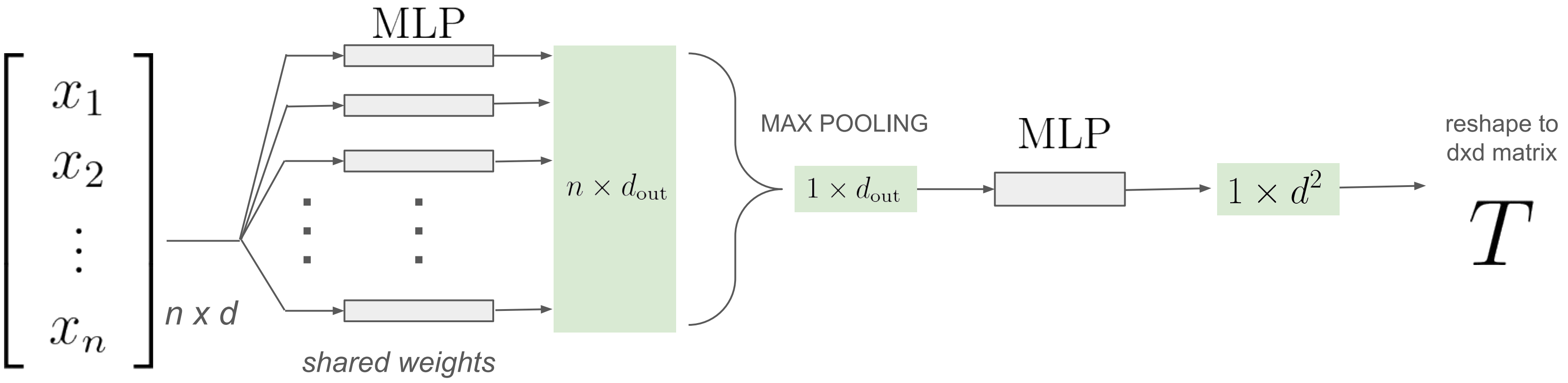
PointNet: Basic Operations

MLP + Max Pooling

shared weights

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

Regress a Transformation Matrix



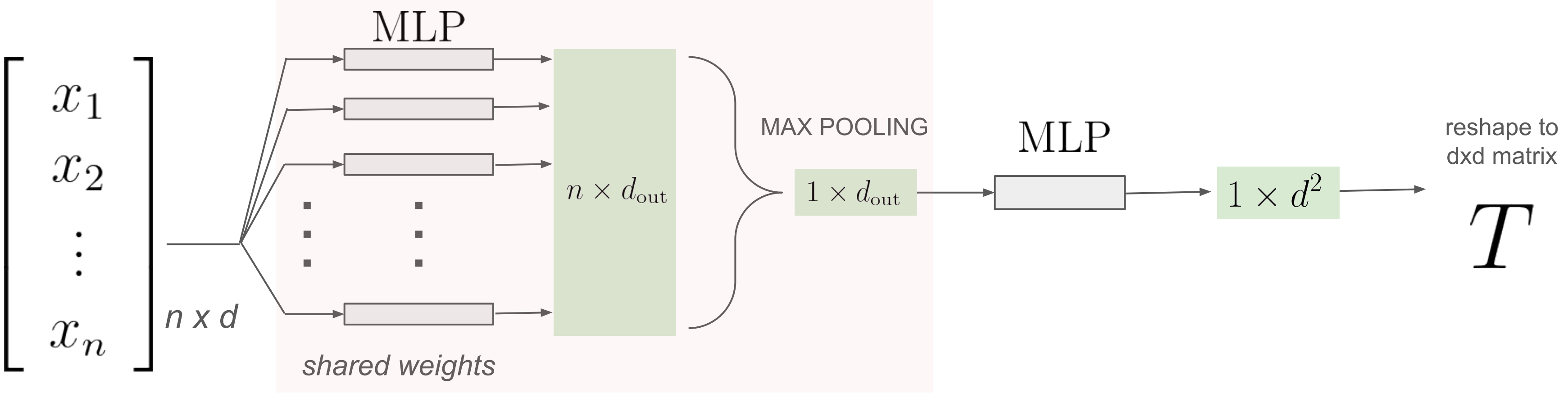
PointNet: Basic Operations

MLP + Max Pooling

shared weights

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

Regress a Transformation Matrix



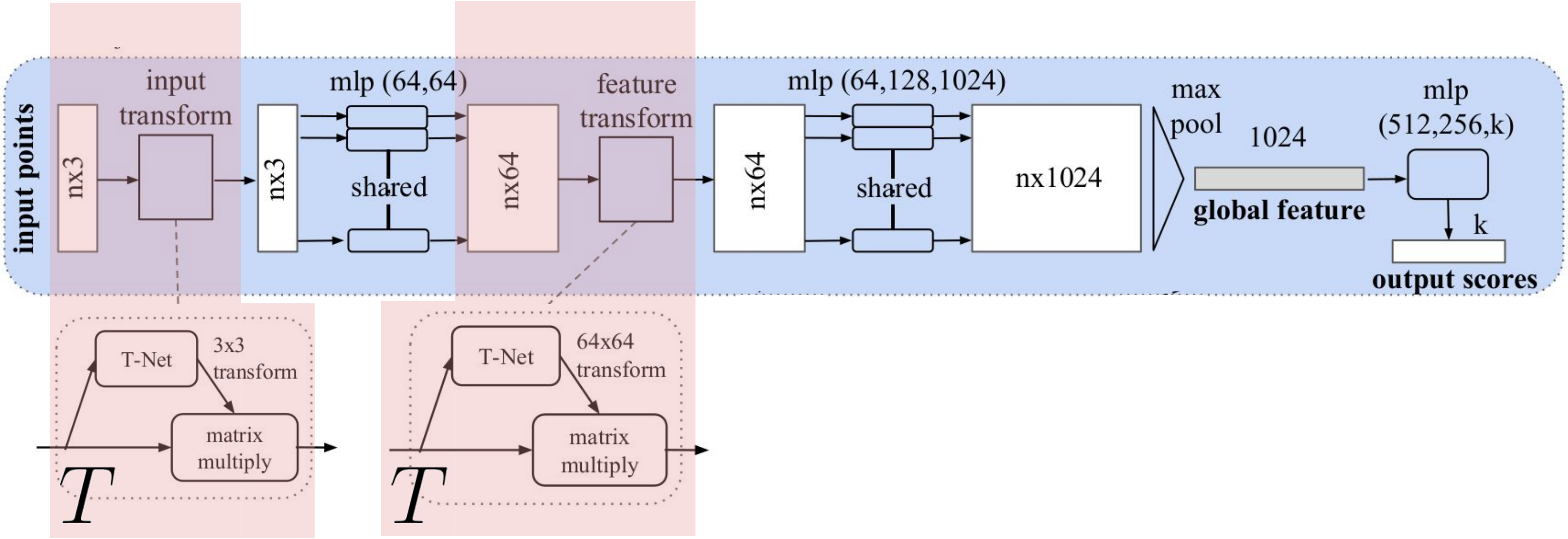
PointNet Architecture

Composition of these two basic operations:

1. MLP + Max Pooling
2. Regress a Transformation Matrix

Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” CVPR 2017

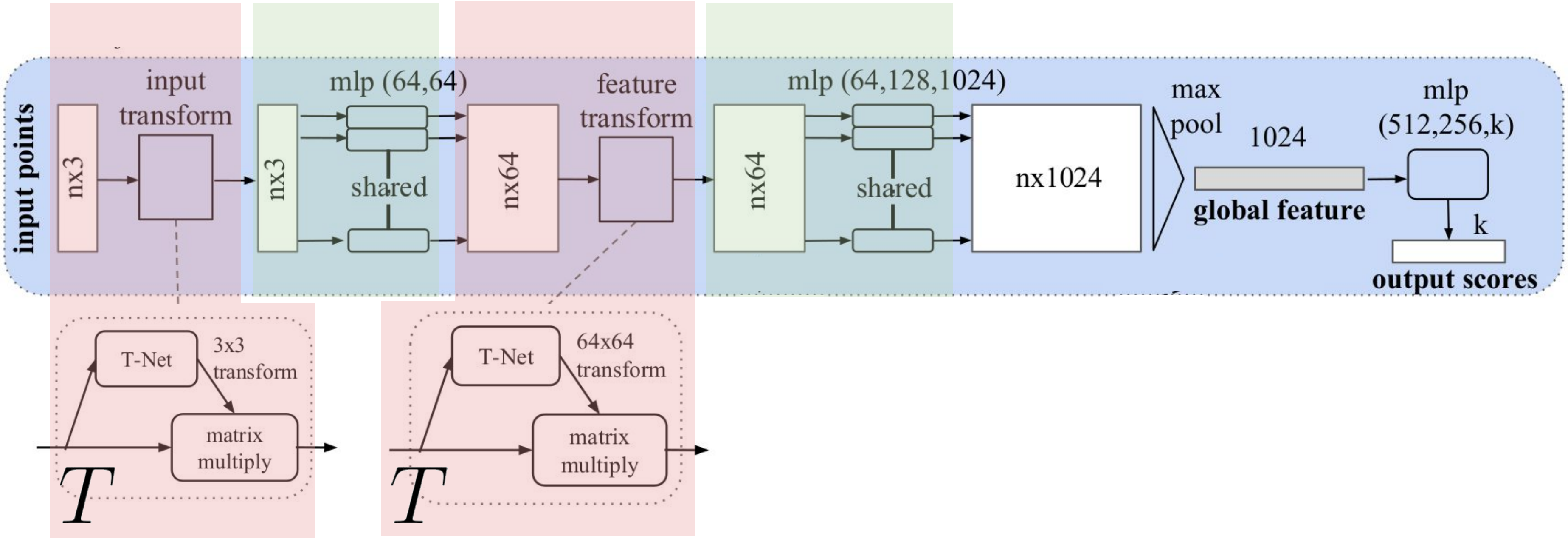
PointNet Architecture



Regresses a transformation matrix and applies it to each input point

Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” CVPR 2017

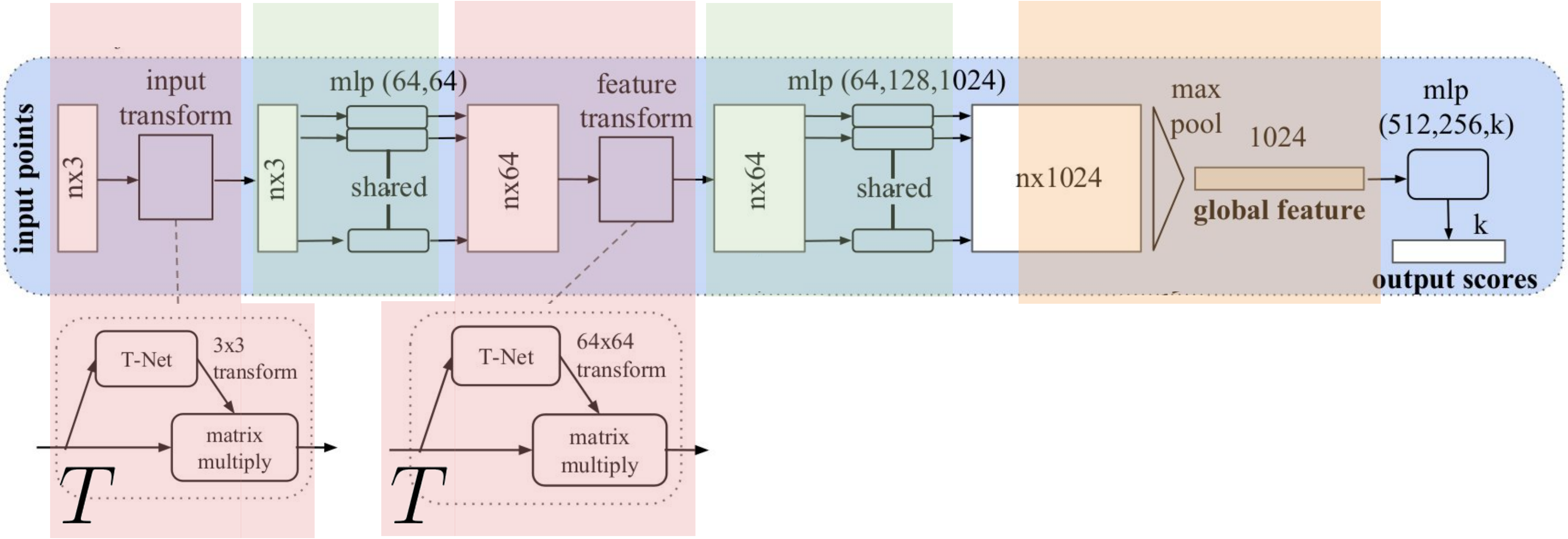
PointNet Architecture



Multi-Layer Perceptron (shared weights) to uplift the dimensions

Qi et al. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" CVPR 2017

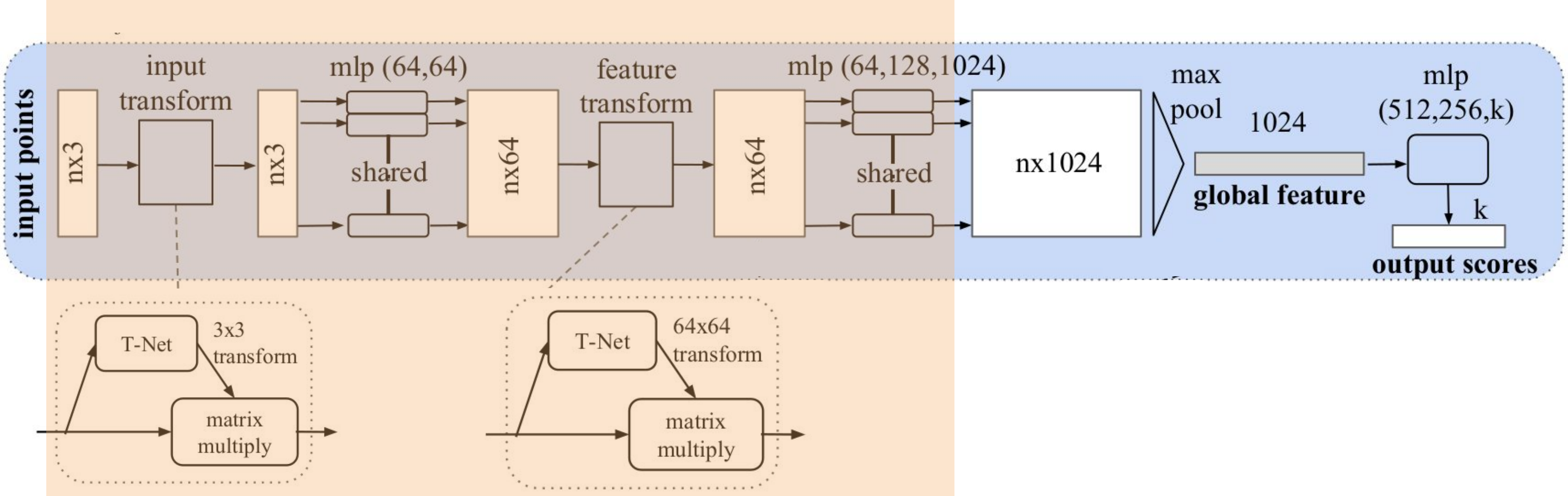
PointNet Architecture



Max Pooling to extract global feature

Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” CVPR 2017

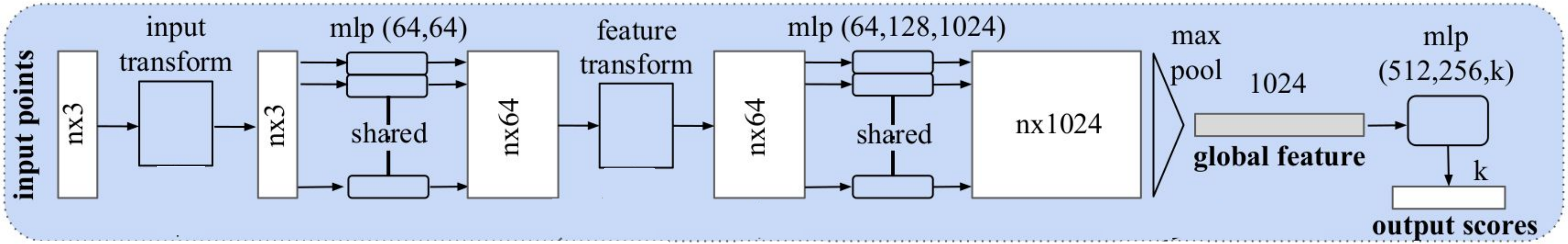
PointNet Architecture



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

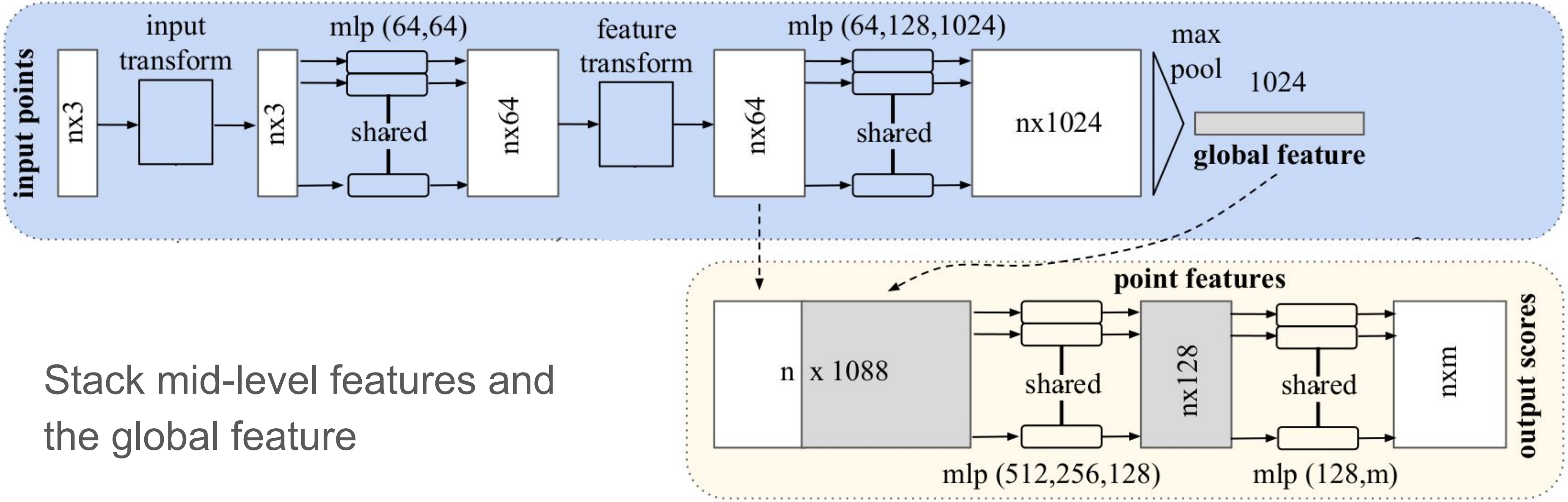
Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” CVPR 2017

PointNet Architecture: Segmentation



Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” CVPR 2017

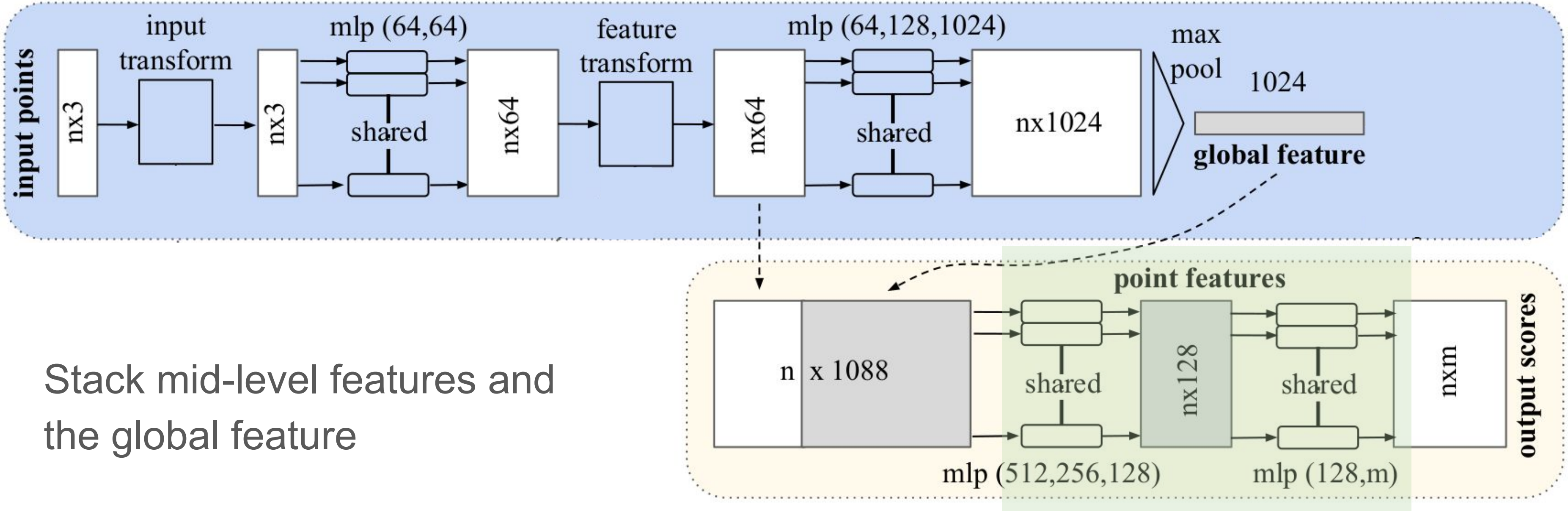
PointNet Architecture: Segmentation



Stack mid-level features and the global feature

Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” CVPR 2017

PointNet Architecture: Segmentation

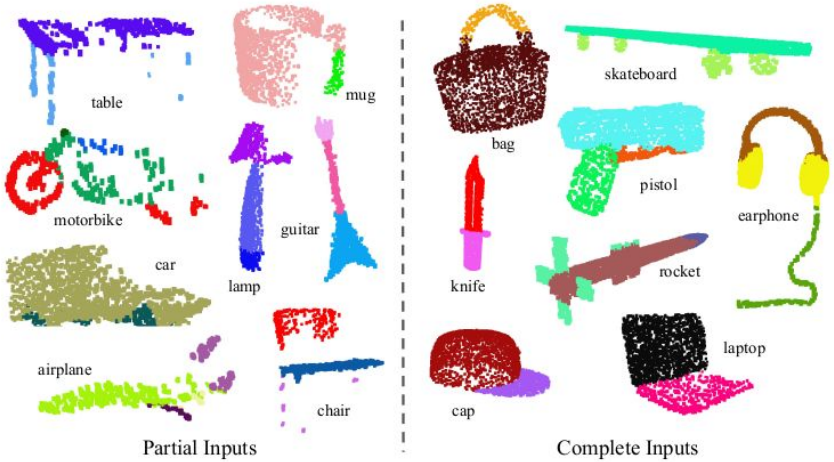


Stack mid-level features and the global feature

Another MLP to extract the final score for each point

Results

Object Part Segmentation

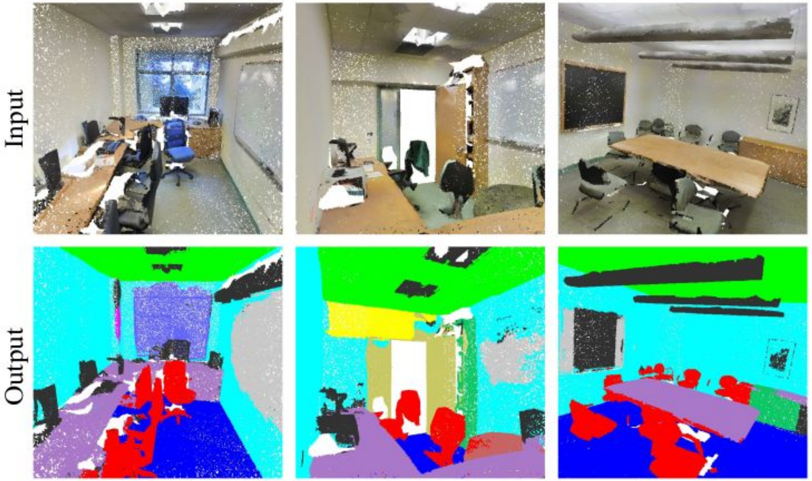


Object Classification

	input	#views	accuracy avg. class	accuracy overall
SPH [11]	mesh	-	68.2	-
3DShapeNets [28]	volume	1	77.3	84.7
VoxNet [17]	volume	12	83.0	85.9
Subvolume [18]	volume	20	86.0	**89.2**
LFD [28]	image	10	75.5	-
MVCNN [23]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

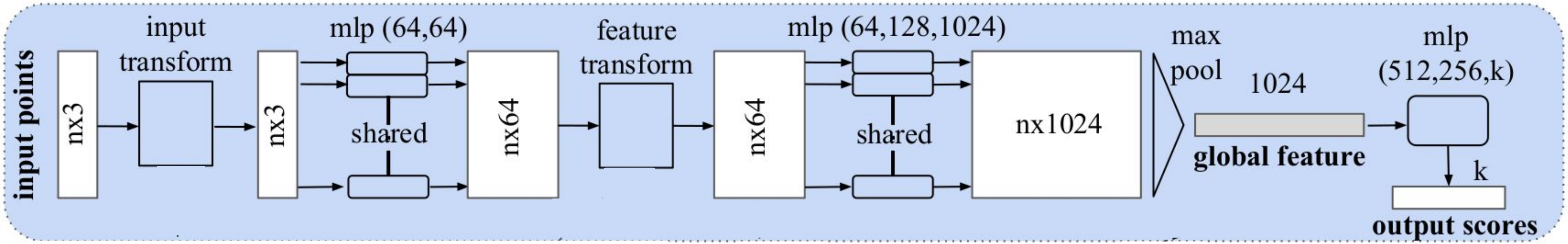
Table 1. **Classification results on ModelNet40.** Our net achieves state-of-the-art among deep nets on 3D input.

State-of-the-art @2017

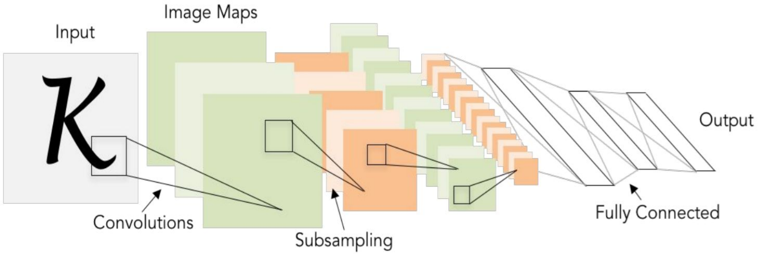


Scene Segmentation

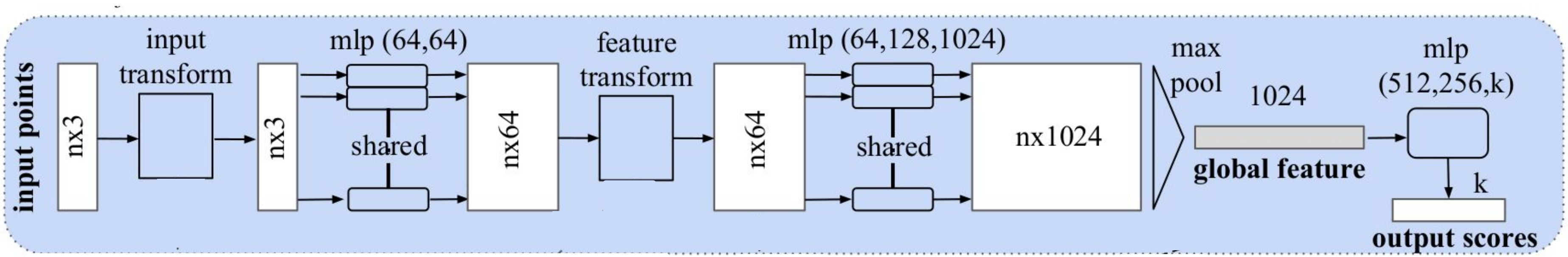
A Limitation of PointNet



Does not extract a sequence of hierarchical features; except a global feature

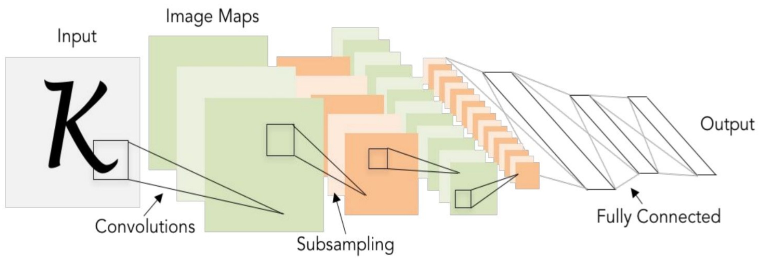


A Limitation of PointNet



Does not extract a sequence of hierarchical features; except a global feature

Does not take into account the local geometry formed by points



Point Clouds

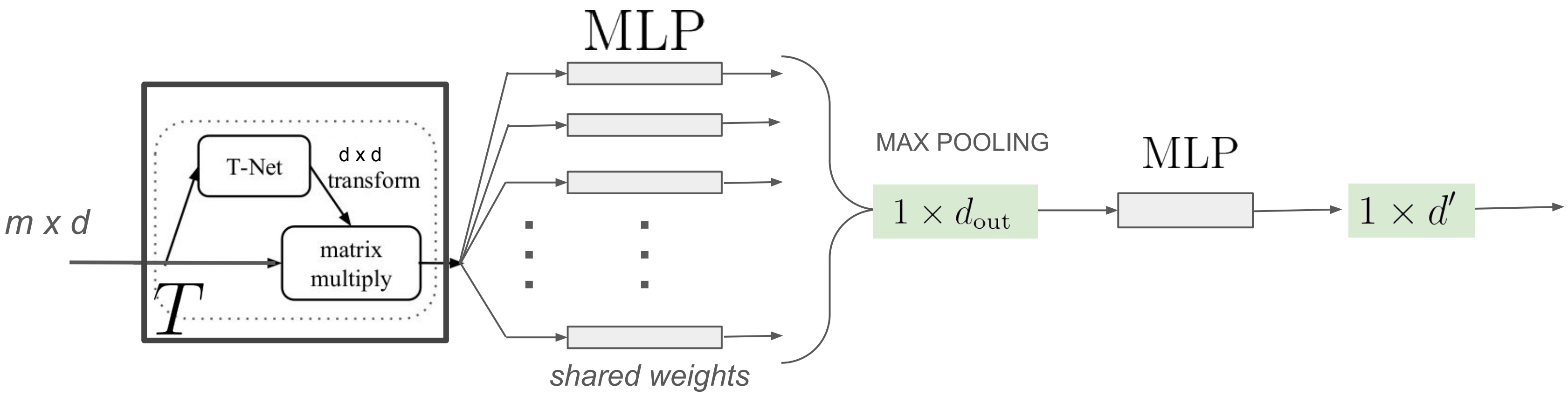
PointNet

PointNet++

PointNet++

Uses PointNet module as a building block

Transforms a set of *m* points to a single point with a feature vector

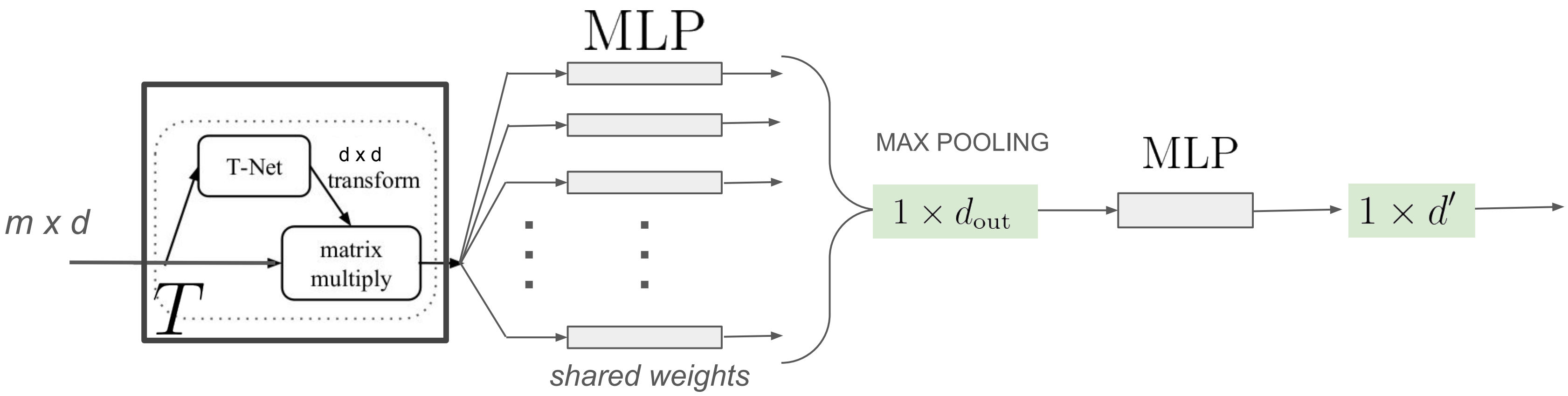


PointNet module

Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

PointNet++

Extracts hierarchical features by recursively applying PointNet module



PointNet module

Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

PointNet++

Sampling

Samples *n'* points using farthest point sampling

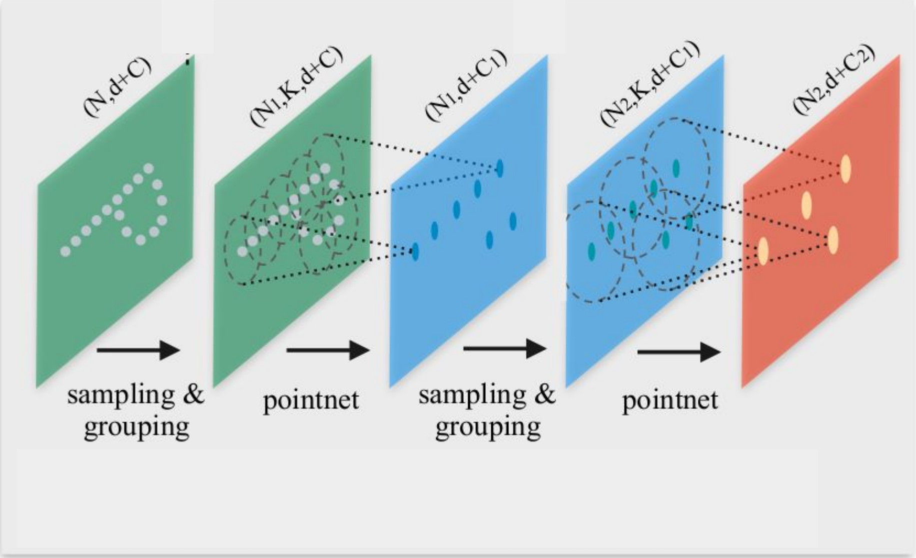
Grouping

For each of the sampled point, selects K points using either

- K-nearest neighbors or
- K points within maximum radius of R

PointNet Layer

Applies PointNet-module to each K-grouping of points and generates a feature vector



Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

PointNet++

Sampling

Samples *n'* points using farthest point sampling

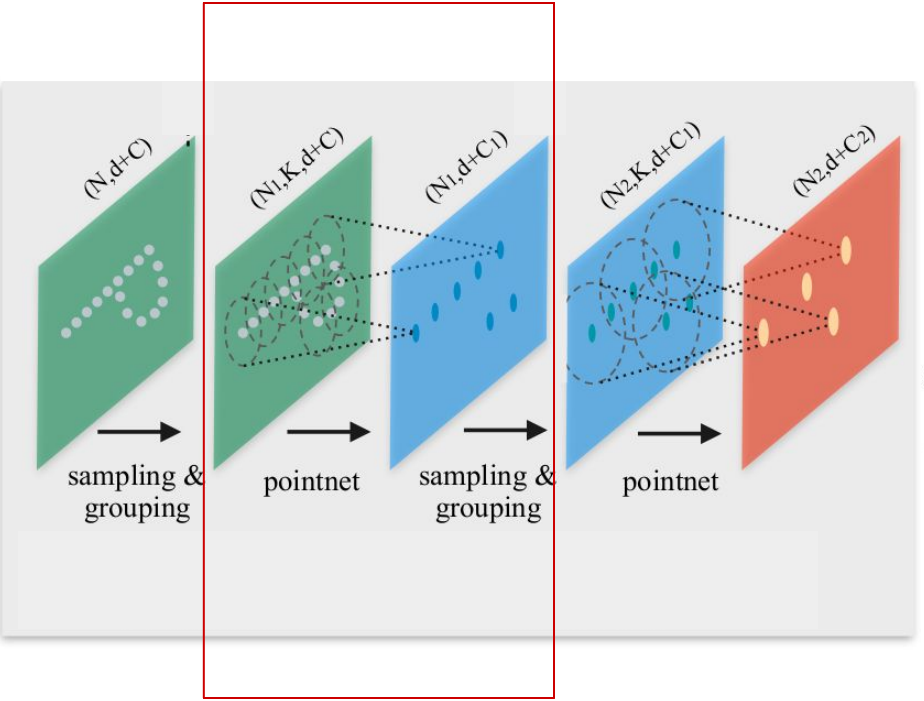
Grouping

For each of the sampled point, selects K points using either

- K-nearest neighbors or
- K points within maximum radius of R

PointNet Layer

Applies PointNet module to each K-grouping of points and generates a feature vector



Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

PointNet++

Looks similar to convolution + pooling?

Sampling

Samples *n'* points using farthest point sampling

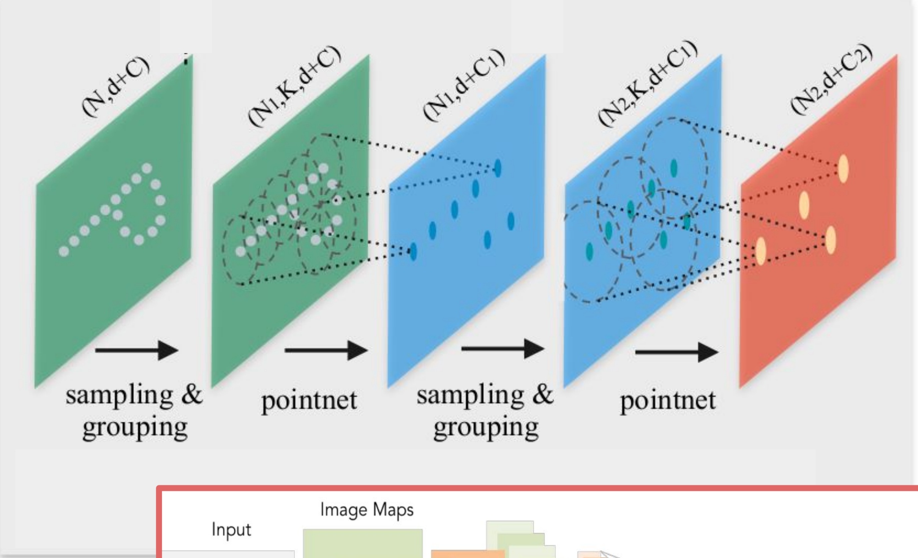
Grouping

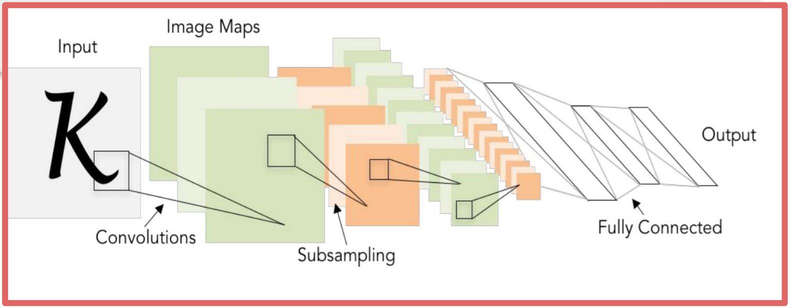
For each of the sampled point, selects K points using either

- K-nearest neighbors or
- K points within maximum radius of R

PointNet Layer

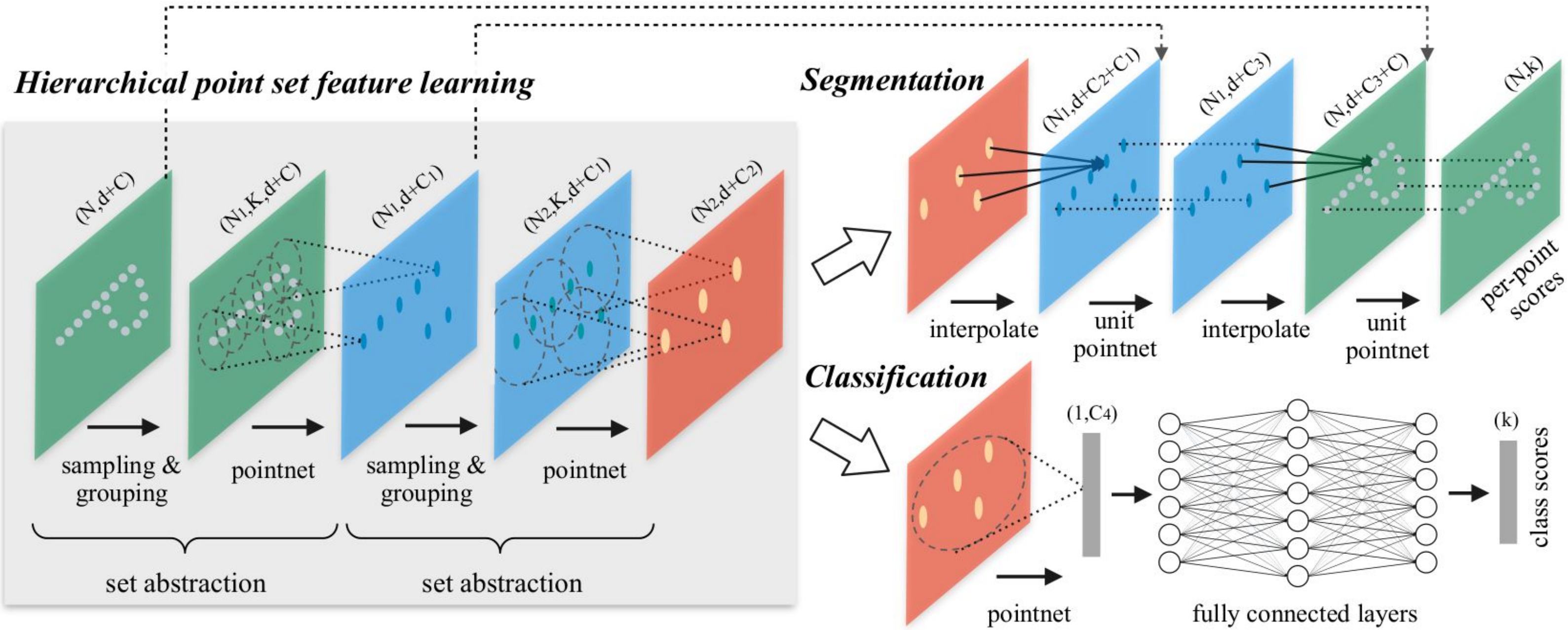
Applies PointNet-module to each K-grouping of points and generates a feature vector





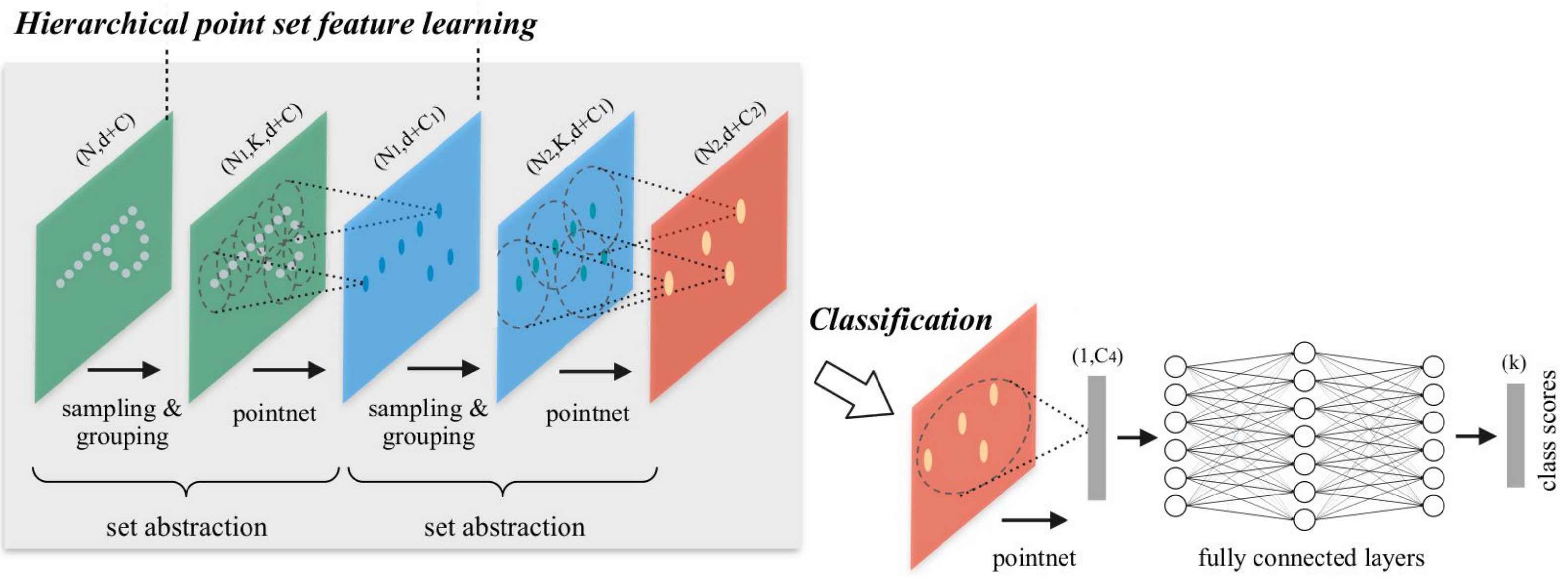
Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

PointNet++ for Classification and Segmentation



Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

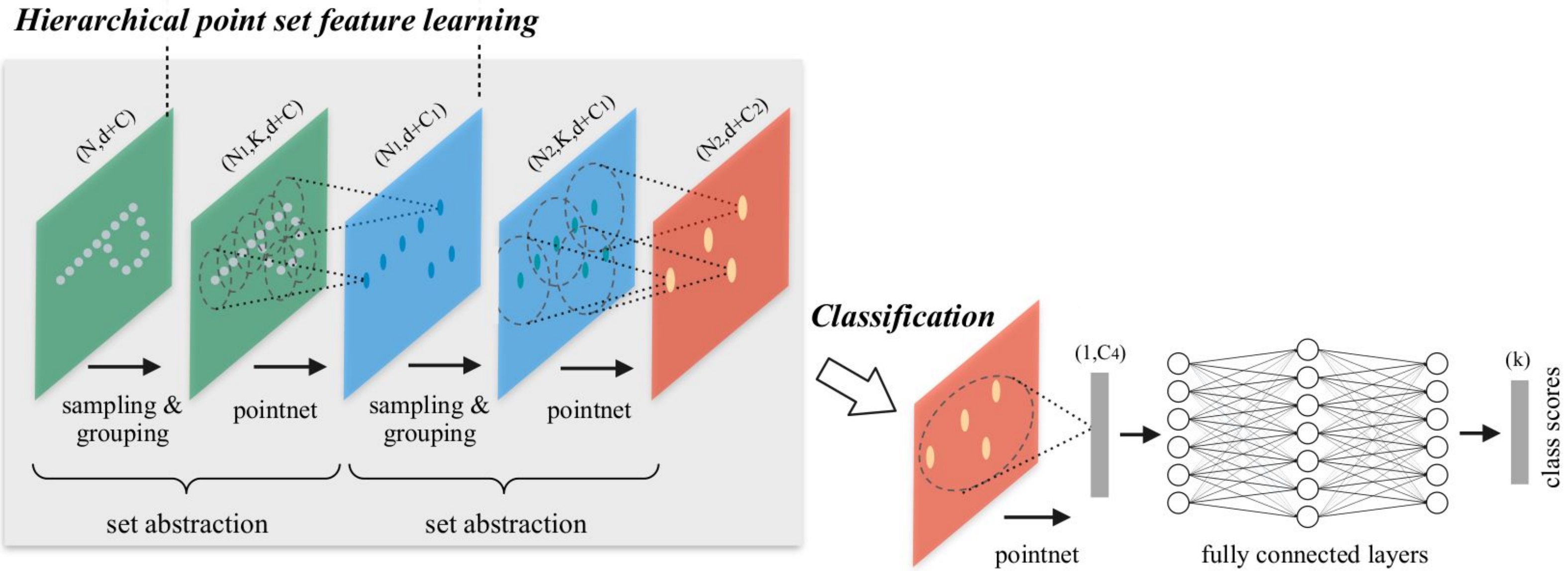
PointNet++ for Classification



Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

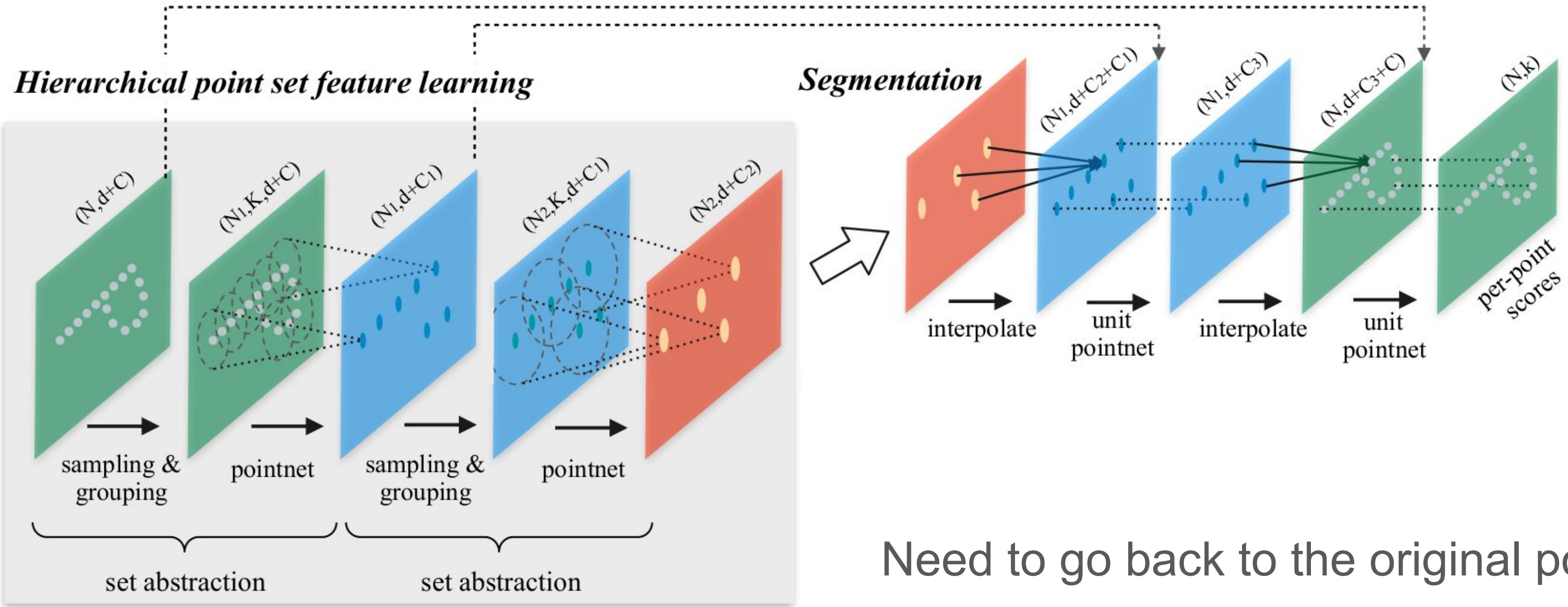
PointNet++ for Classification

Max Pool + MLP on features of the final layer



Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

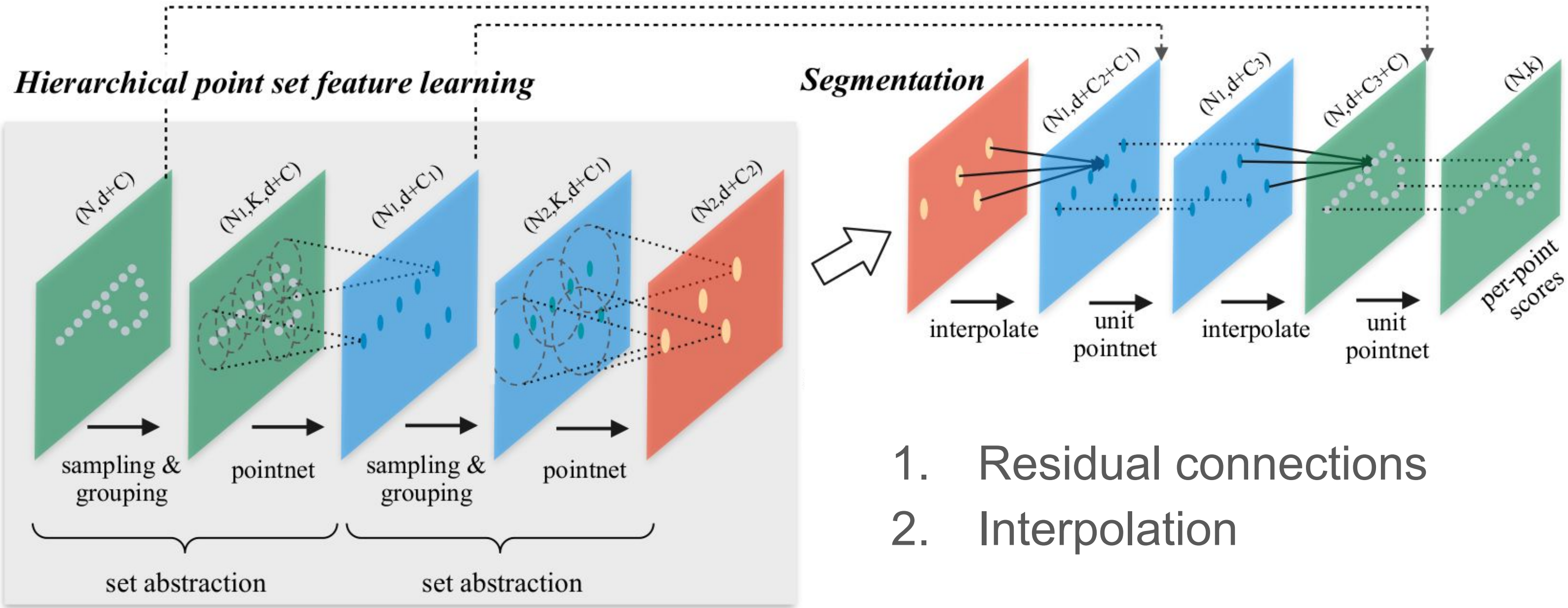
PointNet++ for Segmentation



Need to go back to the original points

Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

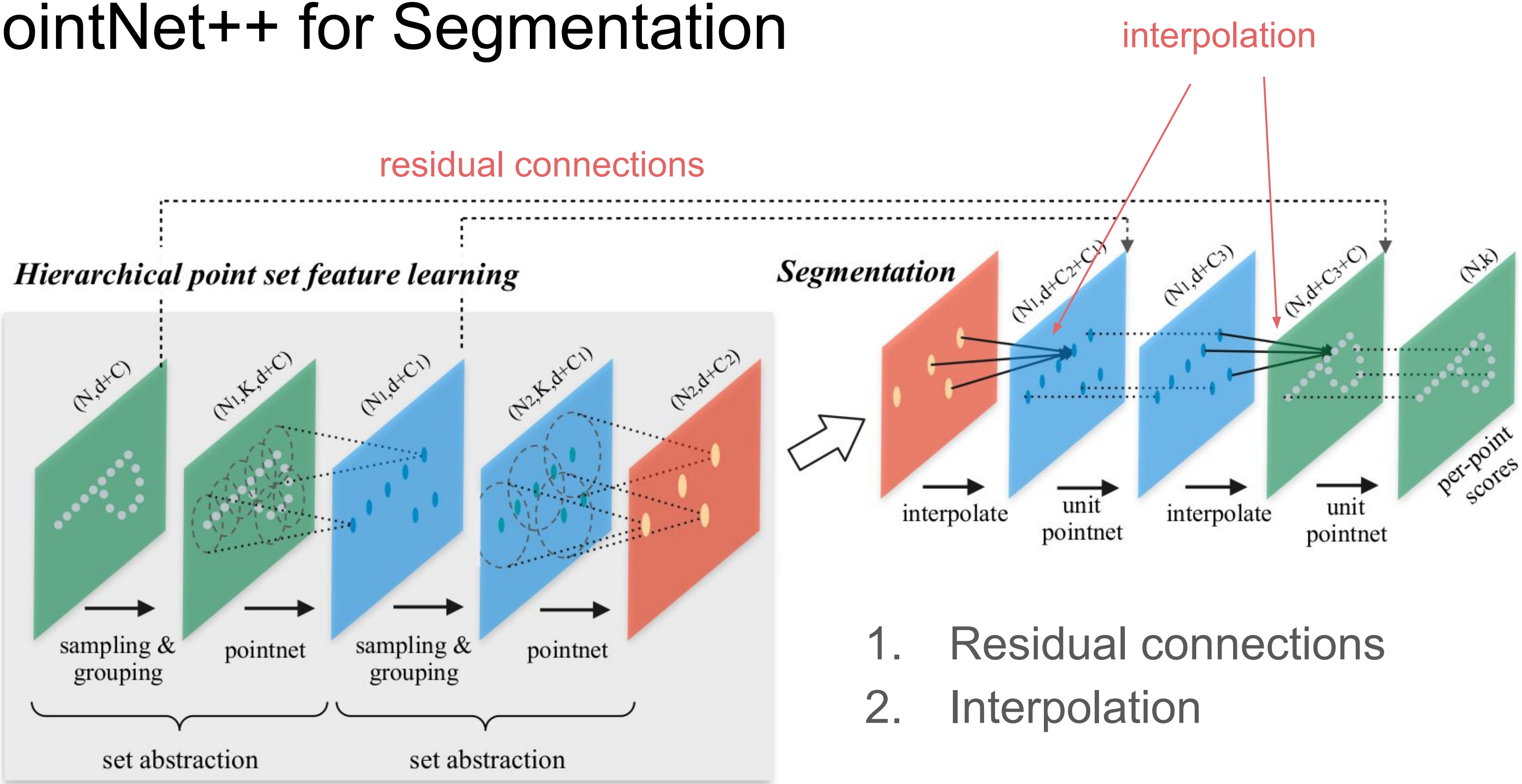
PointNet++ for Segmentation



1. Residual connections
2. Interpolation

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

PointNet++ for Segmentation



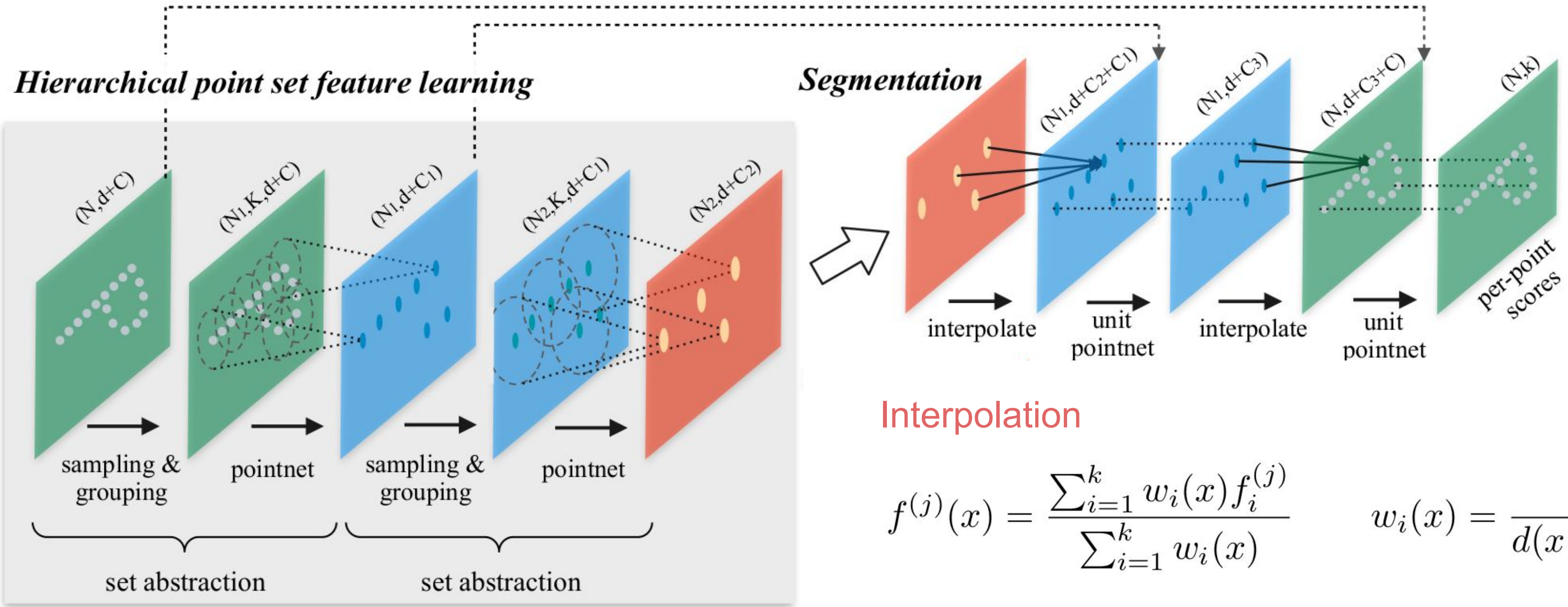
1. Residual connections
2. Interpolation

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

More Details

PointNet++ for Segmentation

These residual connections concatenate features, instead of adding them



Interpolation

$$f^{(j)}(x) = \frac{\sum_{i=1}^{k} w_i(x) f_i^{(j)}}{\sum_{i=1}^{k} w_i(x)} \qquad w_i(x) = \frac{1}{d(x, x_i)^p}$$

$$k = 3, p = 2$$

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

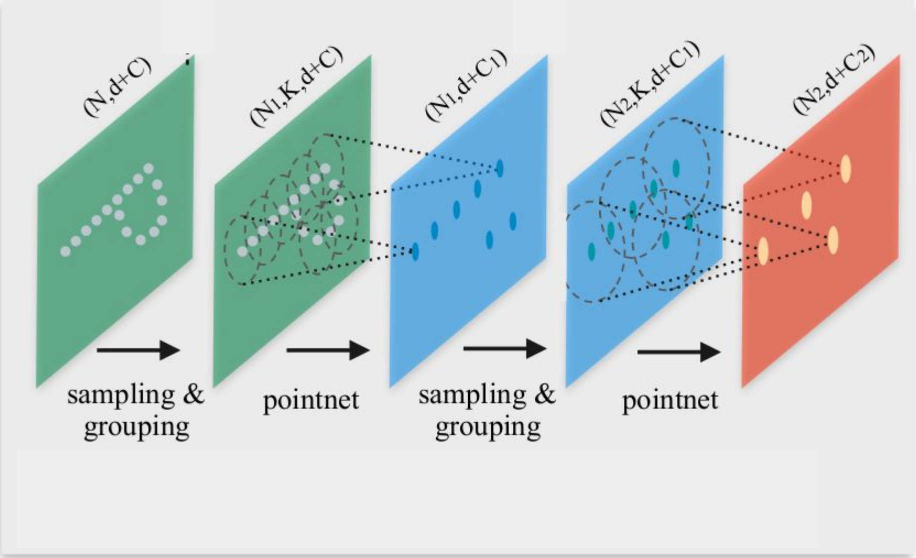
Non-uniform Point Density

PointNet and PointNet ++

implicitly assumes uniform point density

- eg k-nearest neighbors in grouping

Becomes fragile with non-uniform point density



Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

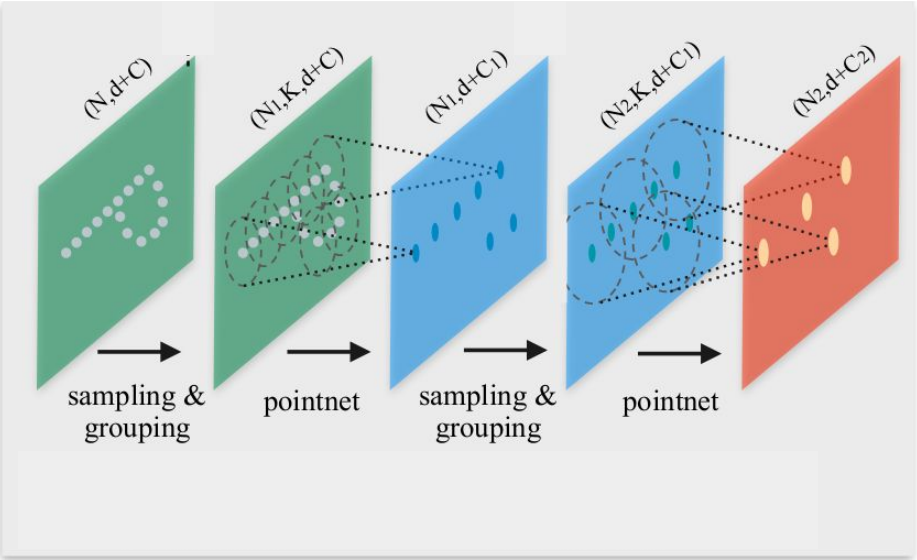
Non-uniform Point Density

PointNet and PointNet ++

implicitly assumes uniform point density

- eg k-nearest neighbors in grouping

Becomes fragile with non-uniform point density

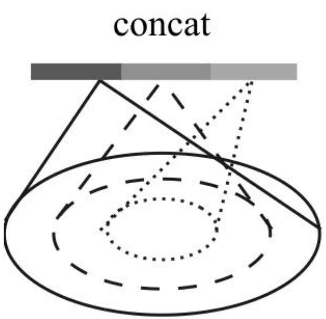


Not an issue on Images or Voxel Grids

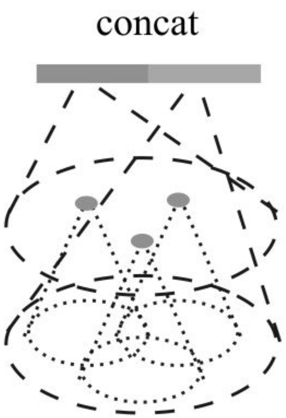
Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

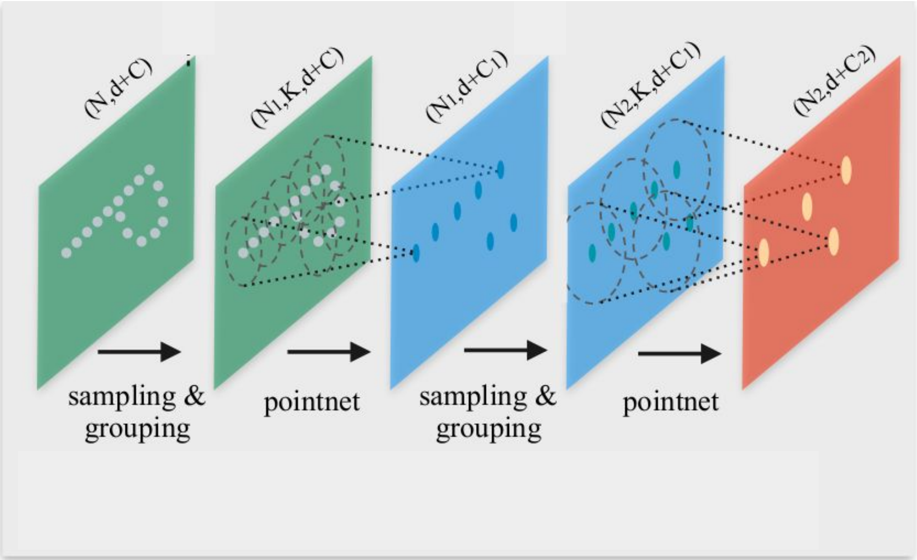
Fix for Non-uniform Point Density

Multi-scale grouping



Multi-resolution grouping



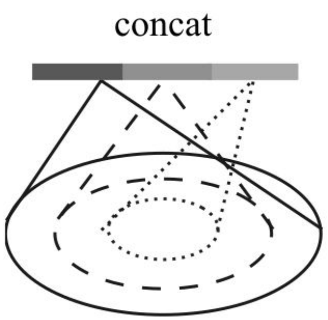


Random Point Dropout at Training

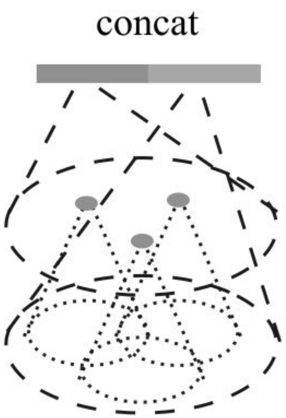
Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

PointNet++

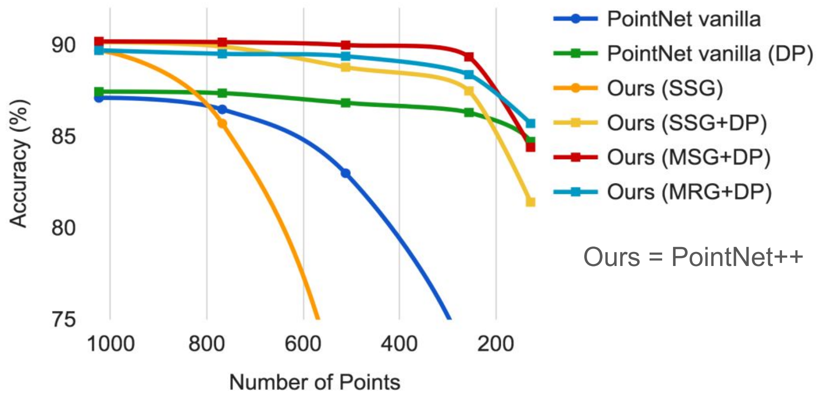
Multi-scale grouping



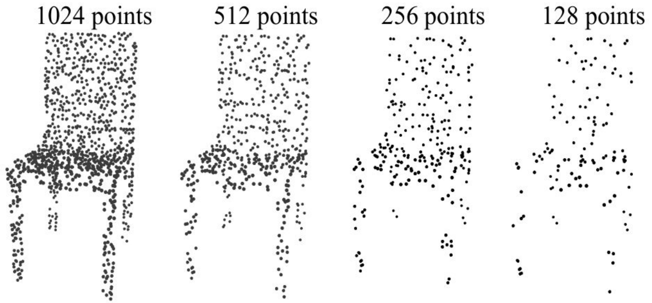
Multi-resolution grouping



Random Point Dropout at Training



Ours = PointNet++

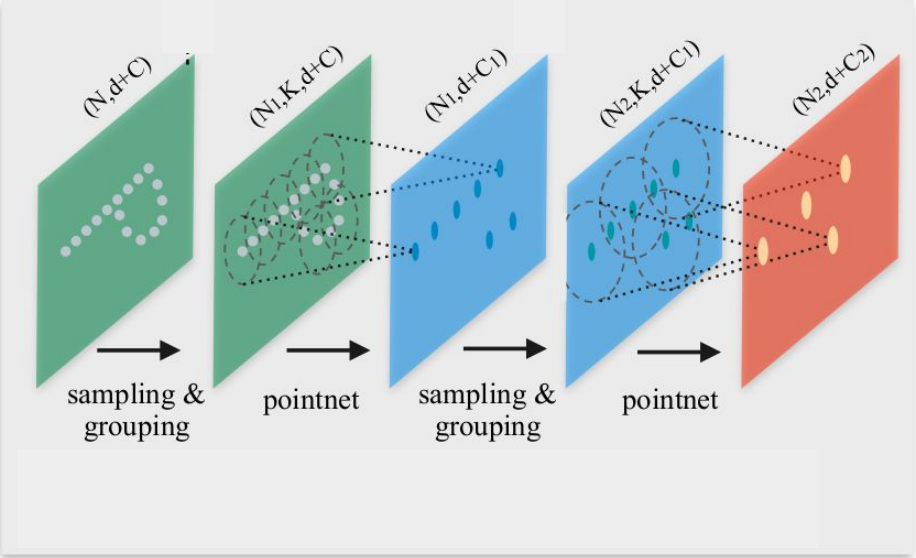


Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

PointNet++

Better Performance than PointNet

Increased Compute Time

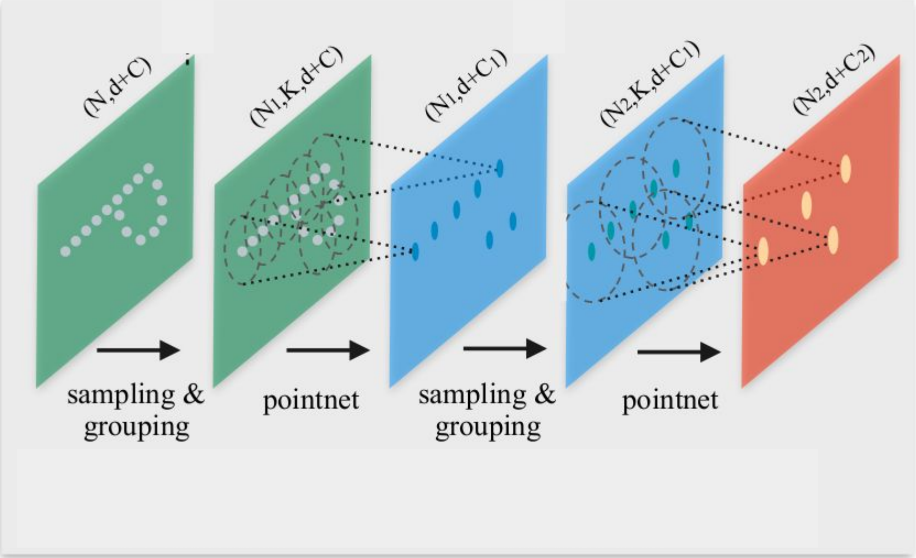


Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

Limitations of PointNet++

Does not take into account the local geometry formed by points

Geometry of hierarchical features are pre-determined



Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space” 2017

Point Clouds

PointNet

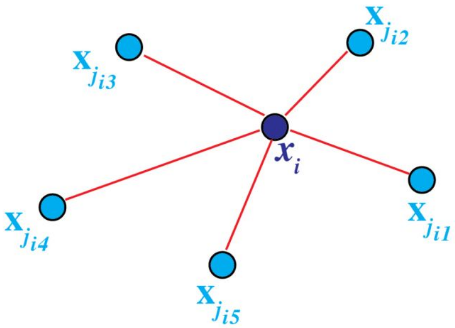
PointNet++

EdgeConv

Wang et al. “Dynamic Graph CNN for Learning on Point Clouds” ACM Trans. Graph 2019

EdgeConv: Basic Idea

Form a local graph by connecting nearby points

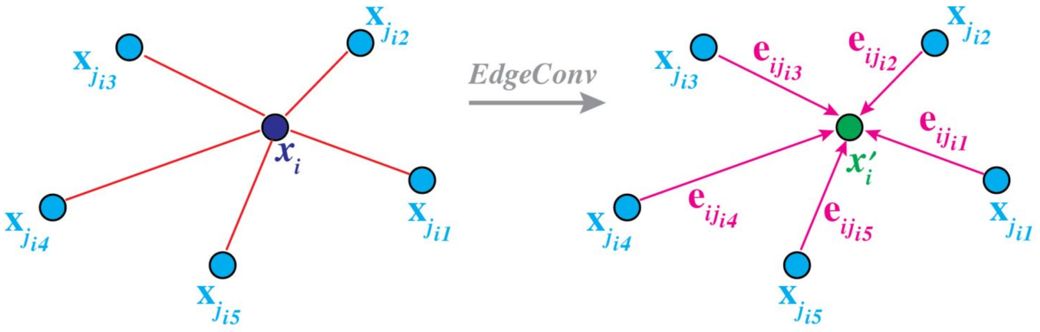


Wang et al. "Dynamic Graph CNN for Learning on Point Clouds" ACM Trans. Graph 2019

EdgeConv: Basic Idea

Form a local graph by connecting nearby points

Apply convolution-like operation on this graph



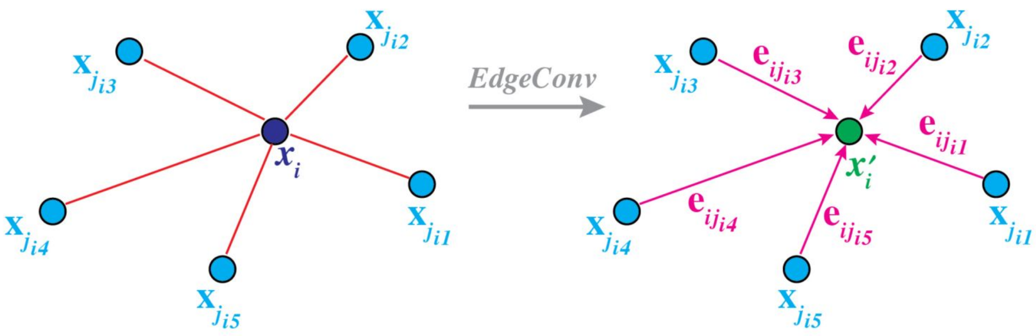
$$x'_i = \square_{j:(i,j)\in E} \; h_\Theta(x_i, x_j)$$

Wang et al. “Dynamic Graph CNN for Learning on Point Clouds” ACM Trans. Graph 2019

EdgeConv: Basic Idea

Form a local graph by connecting nearby points

Apply convolution-like operation on this graph



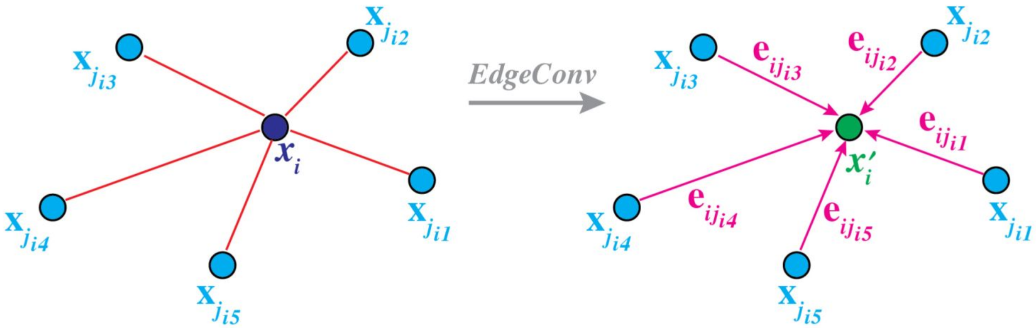
$$x'_i = \square_{j:(i,j)\in E} \; h_\Theta(x_i, x_j)$$

invariant function like max or sum

EdgeConv: Basic Idea

Form a local graph by connecting nearby points

Apply convolution-like operation on this graph



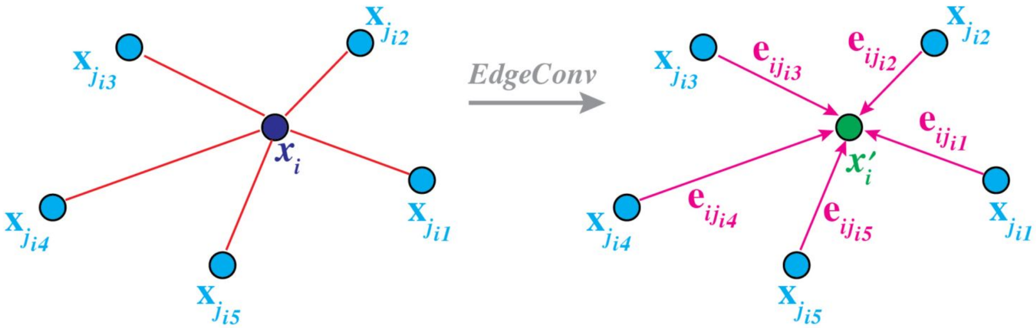
$$x'_i = \square_{j:(i,j)\in E} \; h_\Theta(x_i, x_j)$$

invariant function like max or sum

EdgeConv: Basic Idea

Form a local graph by connecting nearby points

Apply convolution-like operation on this graph



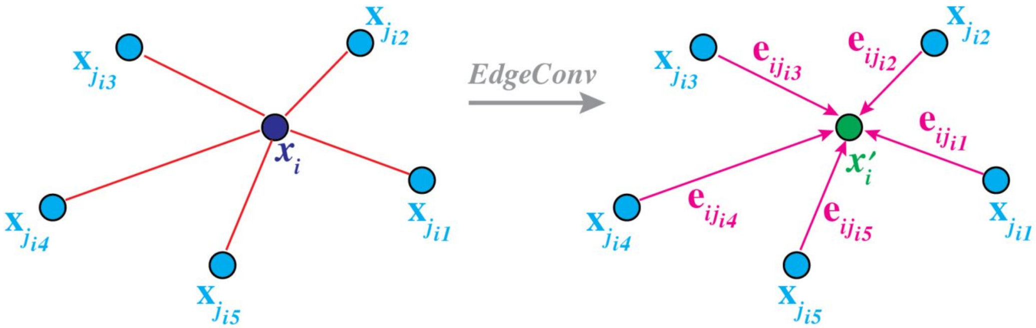
$$x'_i = \square_{j:(i,j)\in E} \; h_\Theta(x_i, x_j)$$

invariant function like max or sum

Nearby: with respect to node feature vectors x_i

EdgeConv: Basic Idea

Form a local graph by connecting nearby points



PointNet++

Connects k-NN from position of points

EdgeConv

Connects k-NN from feature vectors of points

Does this at each layer

EdgeConv Architecture

Step 1: Form a local graph by connecting nearby points with respect to x_i

Step 2: Update feature vectors

$$x_i \leftarrow x'_i = \square_{j:(i,j)\in E} \; h_\Theta(x_i, x_j)$$

EdgeConv Architecture

Step 1: Form a local graph by connecting nearby points with respect to x_i

Step 2: Update feature vectors

$$x_i \leftarrow x_i' = \square_{j:(i,j)\in E} \; h_\Theta(x_i, x_j)$$

iterate

Need to compute a new graph at each stage

EdgeConv Architecture

Step 1: Form a local graph by connecting nearby points with respect to x_i

Step 2: Update feature vectors

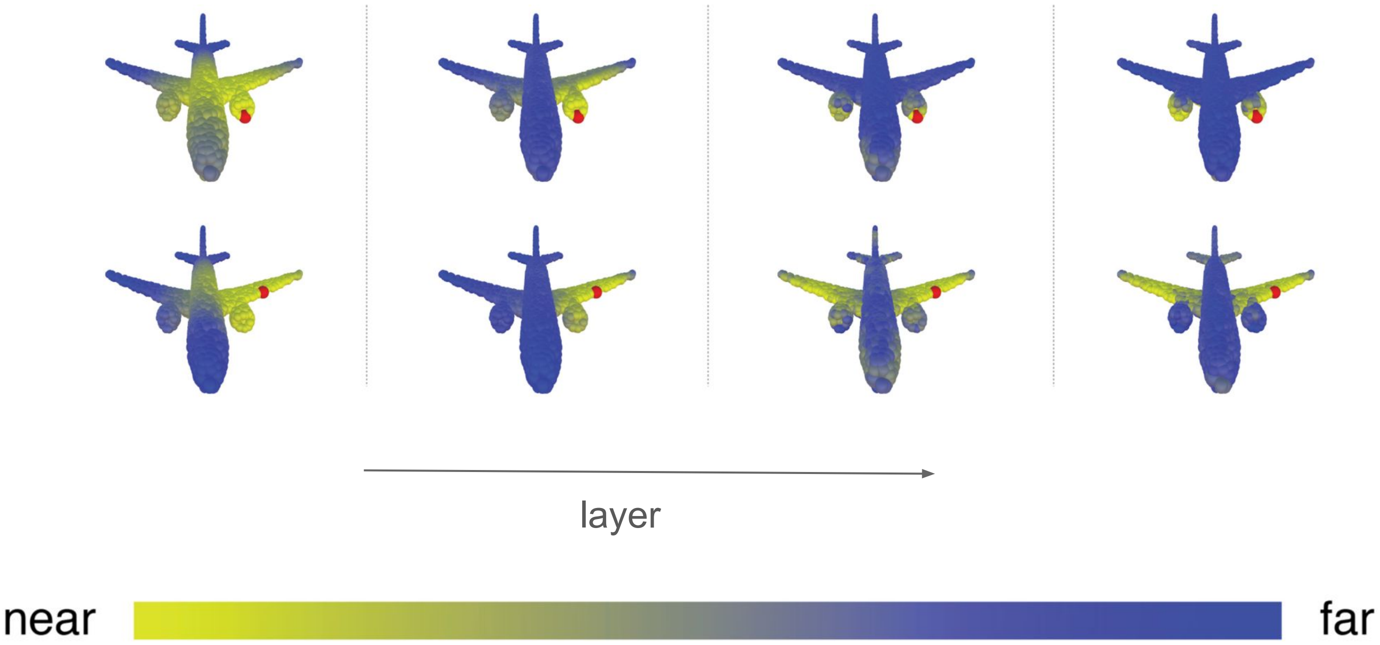
$$x_i \leftarrow x_i' = \square_{j:(i,j)\in E} \; h_\Theta(x_i, x_j)$$

Example

iterate

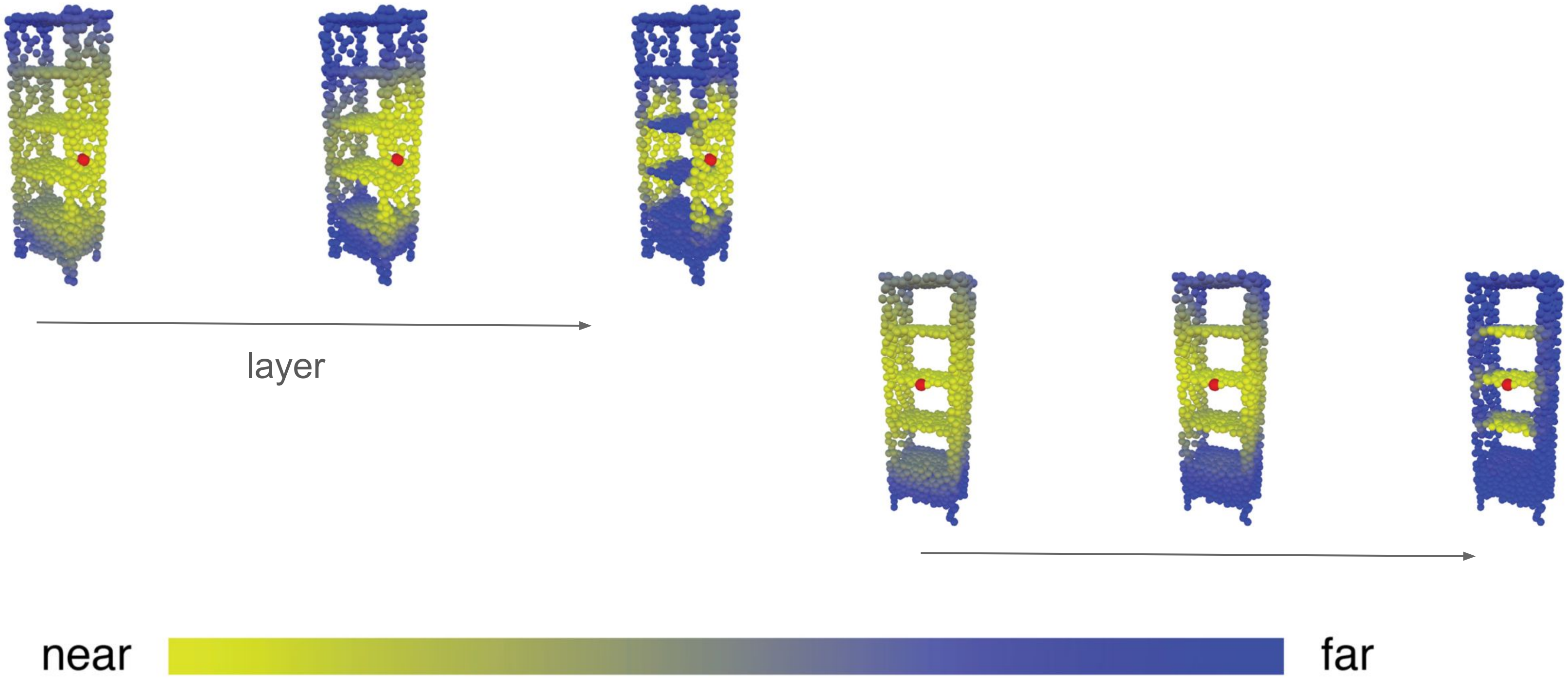
$$h_\Theta(x_i, x_j) = \sigma(\Theta_a \cdot (x_j - x_i) + \Theta_b x_i)$$

Feature Space and Semantically Similar Structures



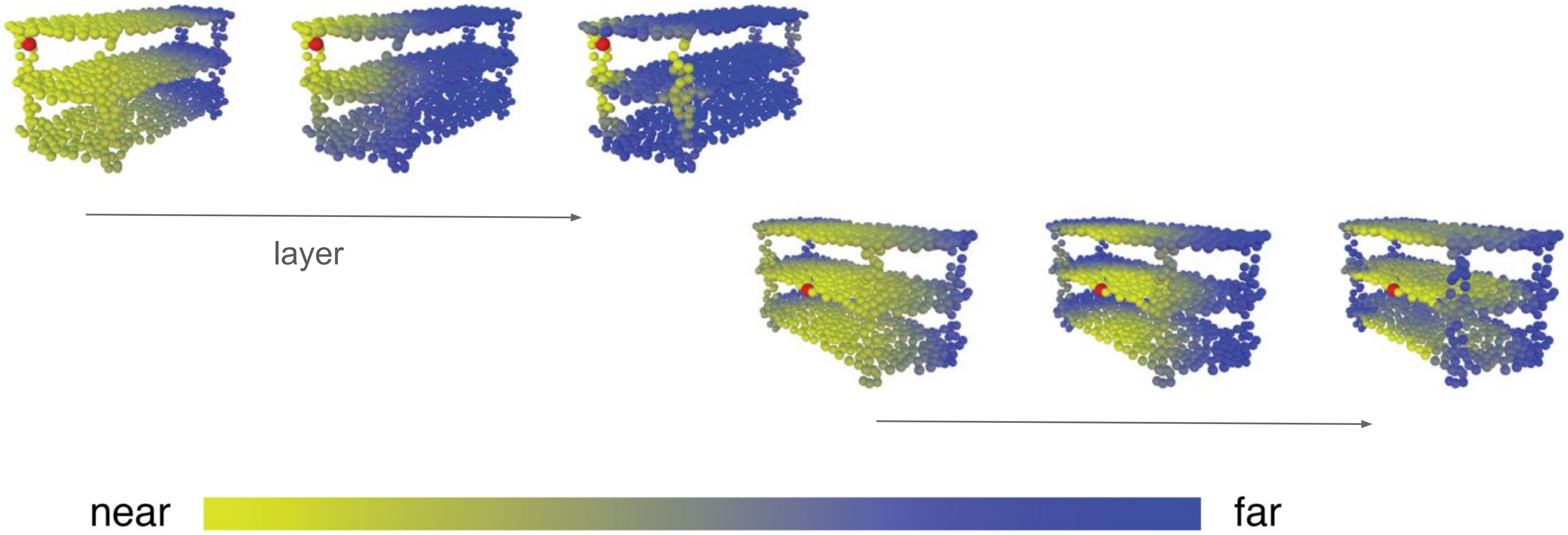
Wang et al. “Dynamic Graph CNN for Learning on Point Clouds” ACM Trans. Graph 2019

Feature Space and Semantically Similar Structures



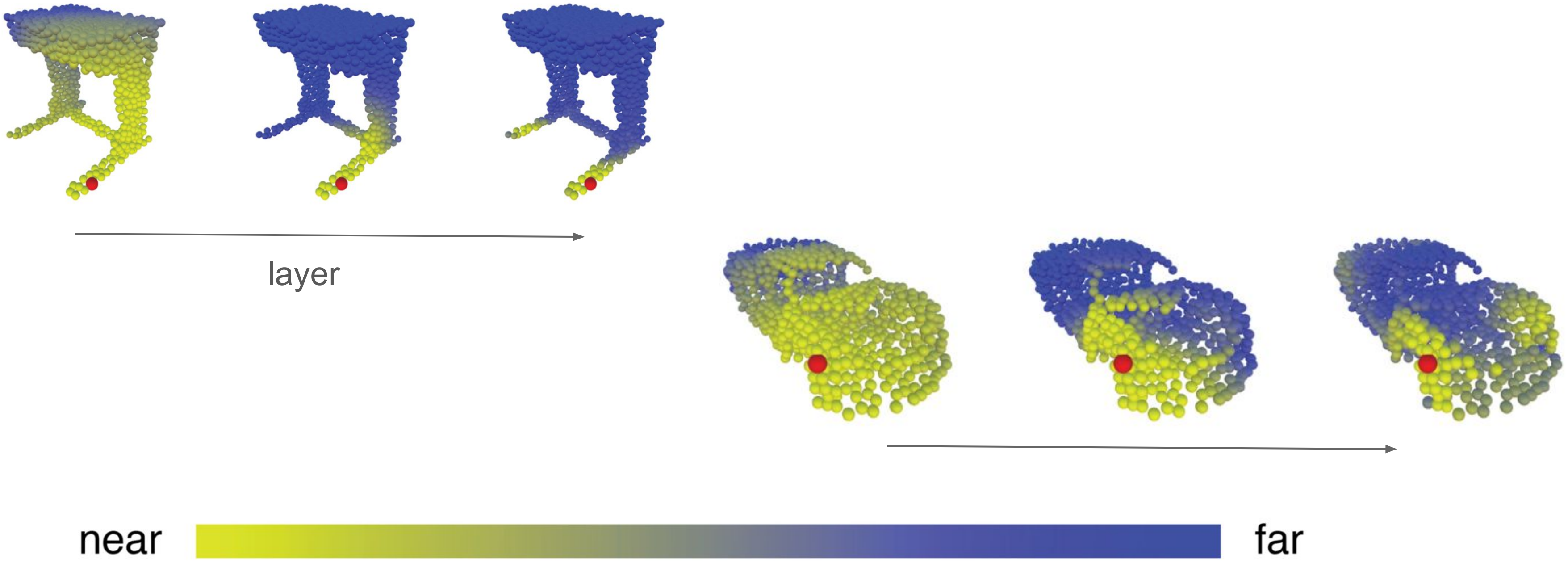
Wang et al. “Dynamic Graph CNN for Learning on Point Clouds” ACM Trans. Graph 2019

Feature Space and Semantically Similar Structures



Wang et al. “Dynamic Graph CNN for Learning on Point Clouds” ACM Trans. Graph 2019

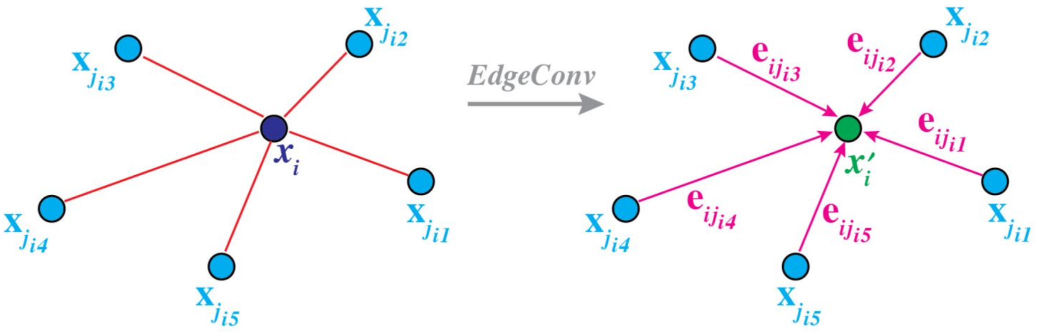
Feature Space and Semantically Similar Structures



Wang et al. “Dynamic Graph CNN for Learning on Point Clouds” ACM Trans. Graph 2019

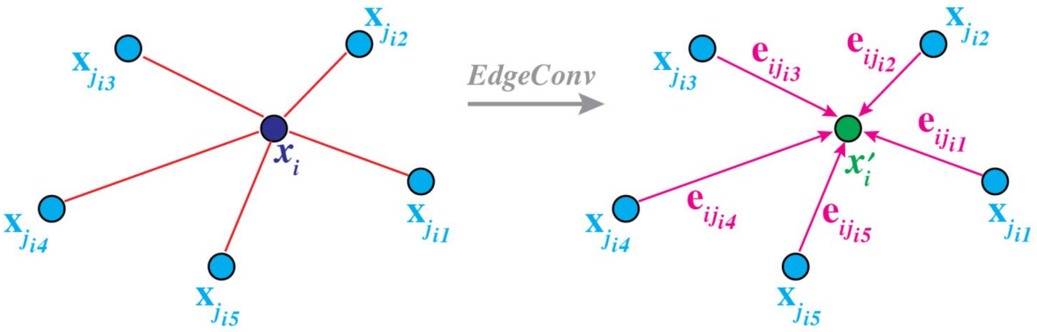
Limitations of EdgeConv

Computationally more expensive than PointNet and PointNet++



Limitations of EdgeConv

Computationally more expensive than PointNet and PointNet++



Is this really a convolution operation?

$$x'_i = \square_{j:(i,j)\in E} \; h_\Theta(x_i, x_j)$$

Point Clouds

PointNet

PointNet++

EdgeConv

KPConv

Point Clouds

PointNet

PointNet++

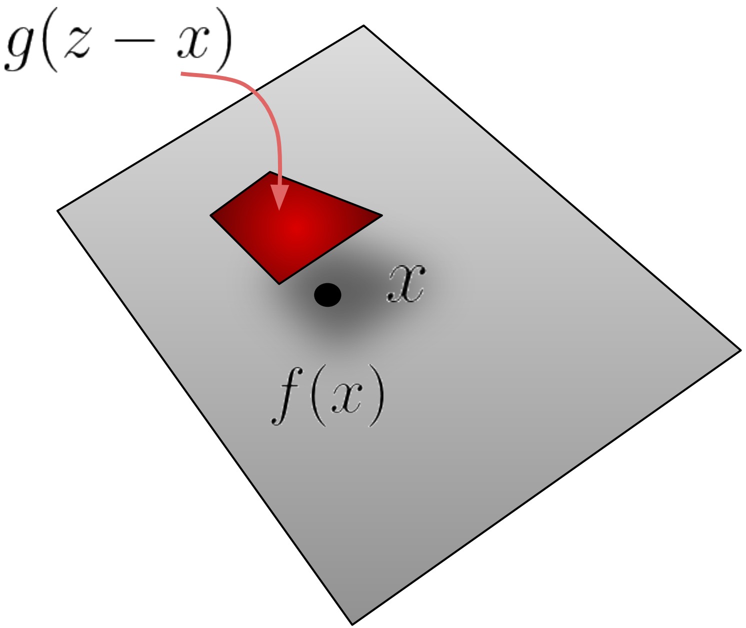
EdgeConv

~~KPConv~~

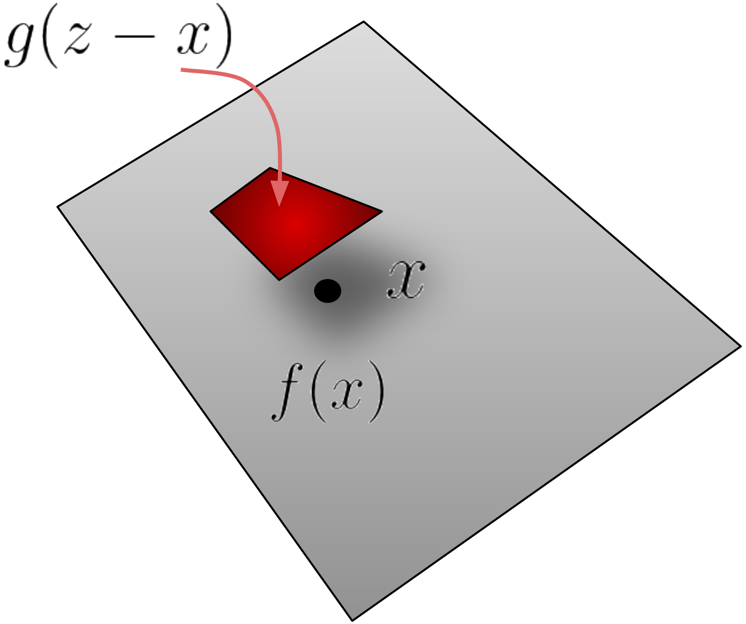
Convolution based architectures for Point Cloud

Convolution

$$(f * g)(x) = \int_{\mathcal{X}} f(z)g(z-x)dz$$



Convolution on Point Clouds?

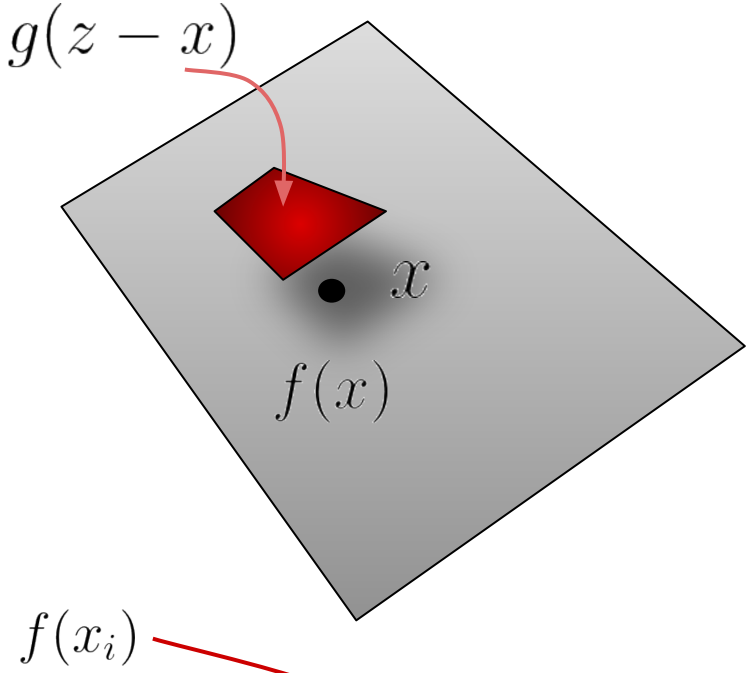


$$(f * g)(x) = \int_{\mathcal{X}} f(z)g(z-x)dz$$

We only have points on $\mathcal{X}$

$$\mathcal{F} = \{(x_i, f_i)\}_i$$

Convolution on Point Clouds?



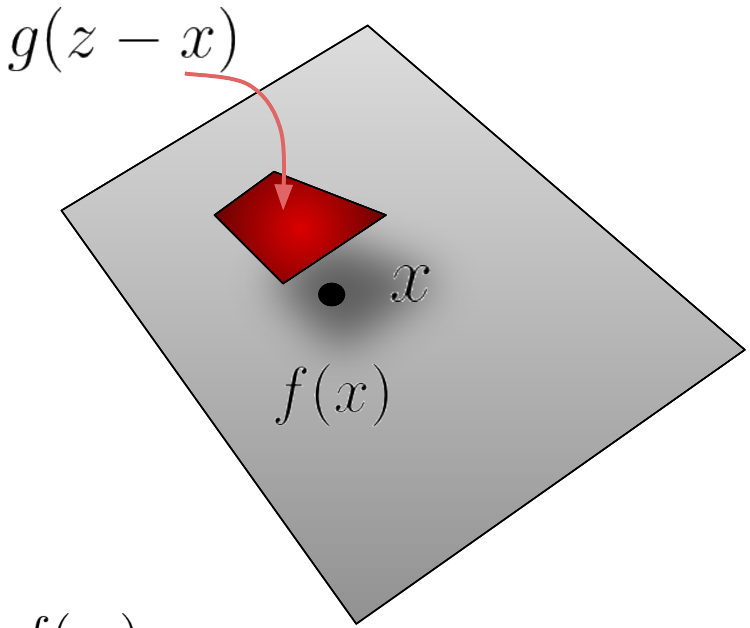
$$(f * g)(x) = \int_{\mathcal{X}} f(z)g(z-x)dz$$

We only have points on $\mathcal{X}$

$$\mathcal{F} = \{(x_i, f_i)\}_i$$

Convolution on Point Clouds?

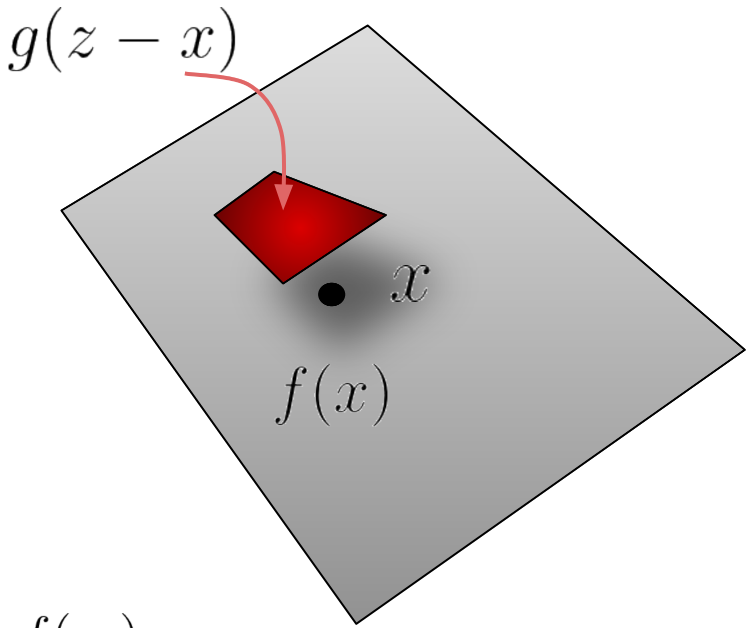
$$(\mathcal{F} * g)(x) = \sum_i f(x_i) g(x_i - x)$$



We only have points on $\mathcal{X}$

$$\mathcal{F} = \{(x_i, f_i)\}_i$$

Convolution on Point Clouds?



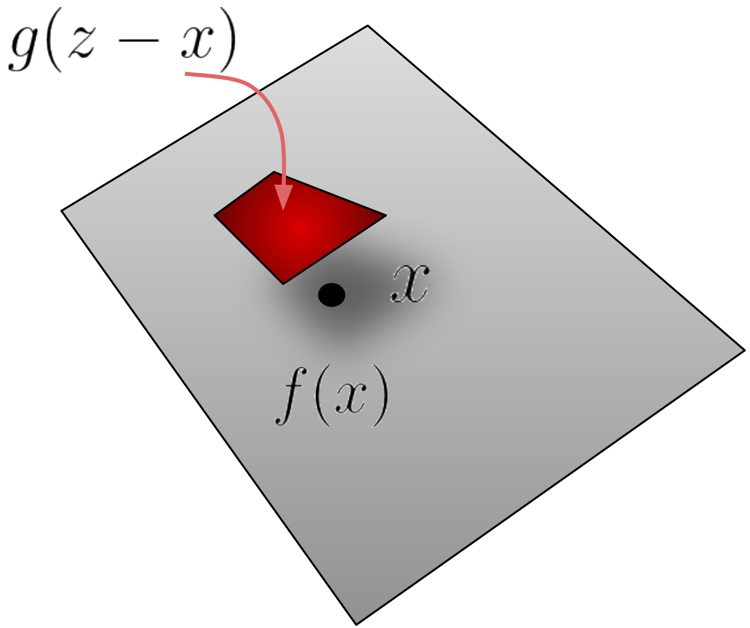
$$(\mathcal{F} * g)(x) = \sum_i f_i \cdot g(x_i - x)$$

$f(x_i)$

We only have points on $\mathcal{X}$

$$\mathcal{F} = \{(x_i, f_i)\}_i$$

Convolution on Point Clouds?



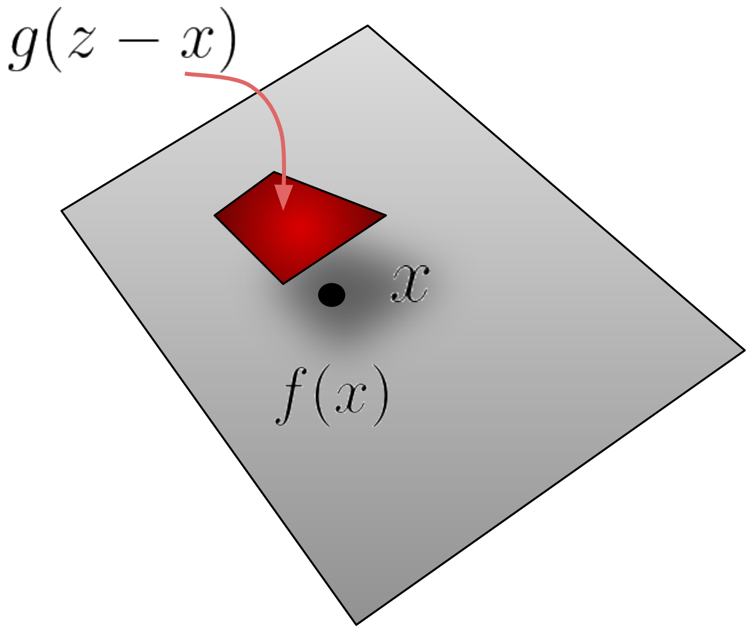
$$(\mathcal{F} * g)(x) = \sum_{i \in N(x)} f_i \cdot g(x_i - x)$$

neighborhood of x

We only have points on $\mathcal{X}$

$$\mathcal{F} = \{(x_i, f_i)\}_i$$

Convolution on Point Clouds



$$(\mathcal{F} * g)(x) = \sum_{i \in N(x)} f_i \cdot g(x_i - x)$$

neighborhood of x

We only have points on $\mathcal{X}$

$$\mathcal{F} = \{(x_i, f_i)\}_i$$

Convolution on Point Clouds

$$(\mathcal{F} * g)(x) = \sum_{i \in N(x)} f_i \cdot g(x_i - x)$$

Point Cloud

$$\mathcal{F} = \{(x_i, f_i)\}_i$$

Convolution on Point Clouds

$$(\mathcal{F} * g)(x) = \sum_{i \in N(x)} f_i \cdot g(x_i - x)$$

Neighborhood

Kernel

Point Cloud

$$\mathcal{F} = \{(x_i, f_i)\}_i$$

Convolution on Point Clouds

$$(\mathcal{F} * g)(x) = \underbrace{\sum_{i \in N(x)}}_{\text{Neighborhood}} f_i \cdot \underbrace{g(x_i - x)}_{\text{Kernel}}$$

Point Cloud

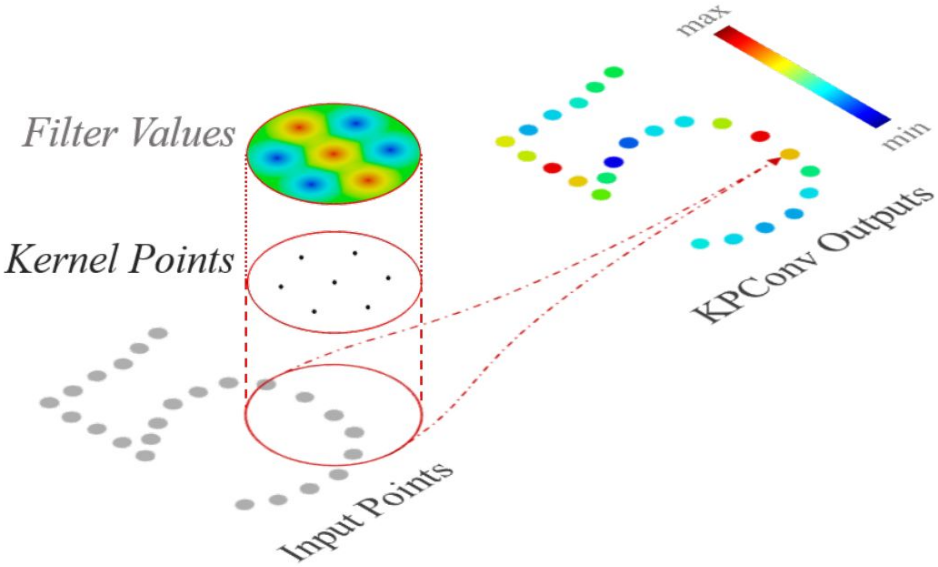
$$\mathcal{F} = \{(x_i, f_i)\}_i$$

Many choices of kernel functions in the literature.

Kernel Point Convolution (KPConv)

$$g(z) = \sum_{1 \le k \le K} h(z, z_k) W_k$$

A specific choice of kernel function

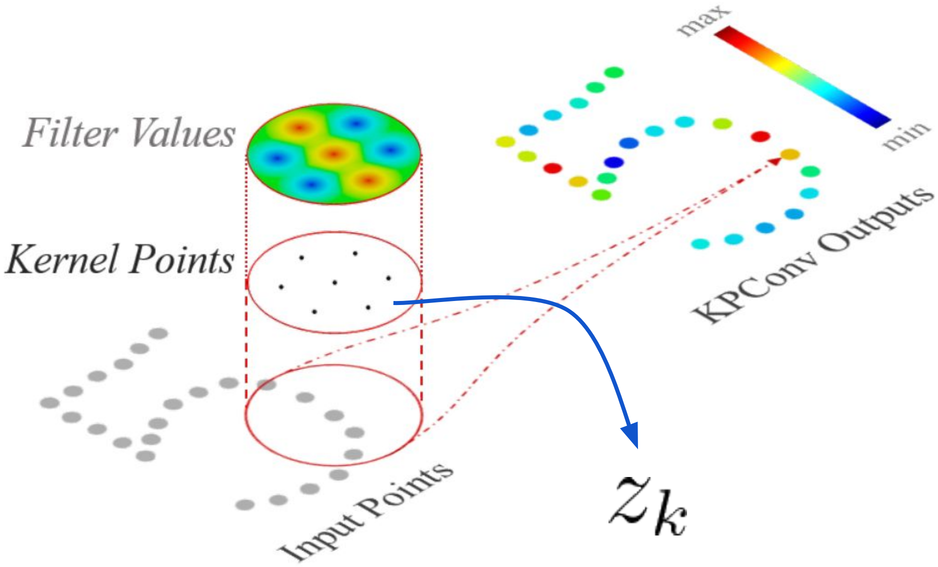


Thomas et al. “KPConv: Flexible and Deformable Convolution for Point Clouds” 2019

Kernel Point Convolution (KPConv)

$$g(z) = \sum_{1 \leq k \leq K} h(z, z_k) W_k$$

A specific choice of kernel function



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

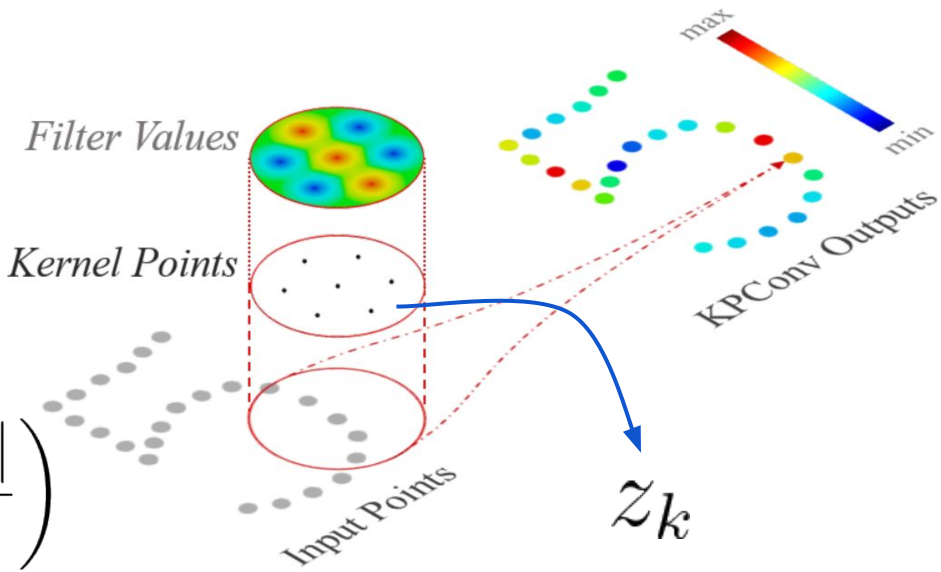
More Details

Kernel Point Convolution (KPConv)

$$g(z) = \sum_{1 \le k \le K} h(z, z_k) W_k$$

where

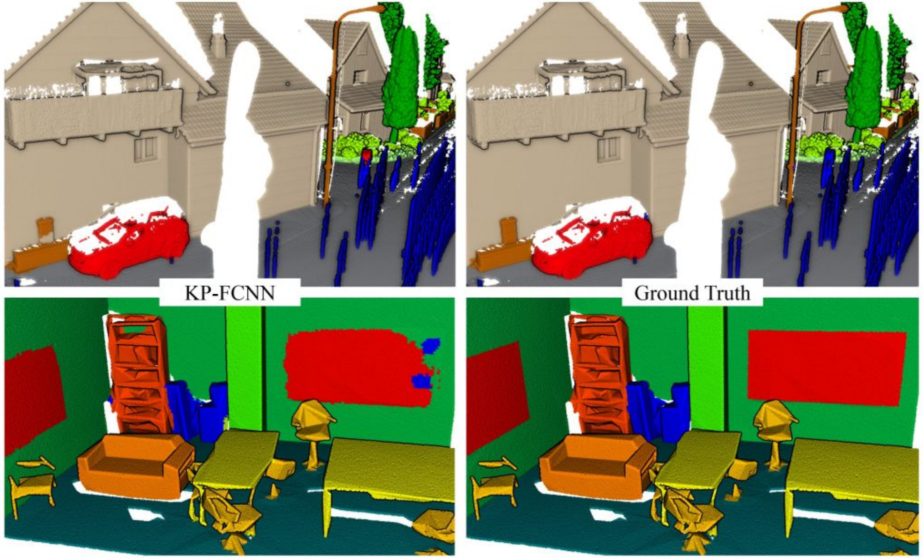
$$h(z, z_k) = \max\left(0, 1 - \frac{||z - z_k||}{\sigma}\right)$$



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

KPConv Performance

	ModelNet40	ShapeNetPart	
Methods	OA	mcIoU	mIoU
SPLATNet [34]	-	83.7	85.4
SGPN [42]	-	82.8	85.8
3DmFV-Net [9]	91.6	81.0	84.3
SynSpecCNN [48]	-	82.0	84.7
RSNet [15]	-	81.4	84.9
SpecGCN [40]	91.5	-	85.4
PointNet++ [27]	90.7	81.9	85.1
SO-Net [19]	90.9	81.0	84.9
PCNN by Ext [2]	92.3	81.8	85.1
SpiderCNN [45]	90.5	82.4	85.3
MCConv [13]	90.9	-	85.9
FlexConv [10]	90.2	84.7	85.0
PointCNN [20]	92.2	84.6	86.1
DGCNN [43]	92.2	85.0	84.7
SubSparseCNN [9]	-	83.3	86.0
KPConv *rigid*	**92.9**	85.0	86.2
KPConv *deform*	92.7	**85.1**	**86.4**

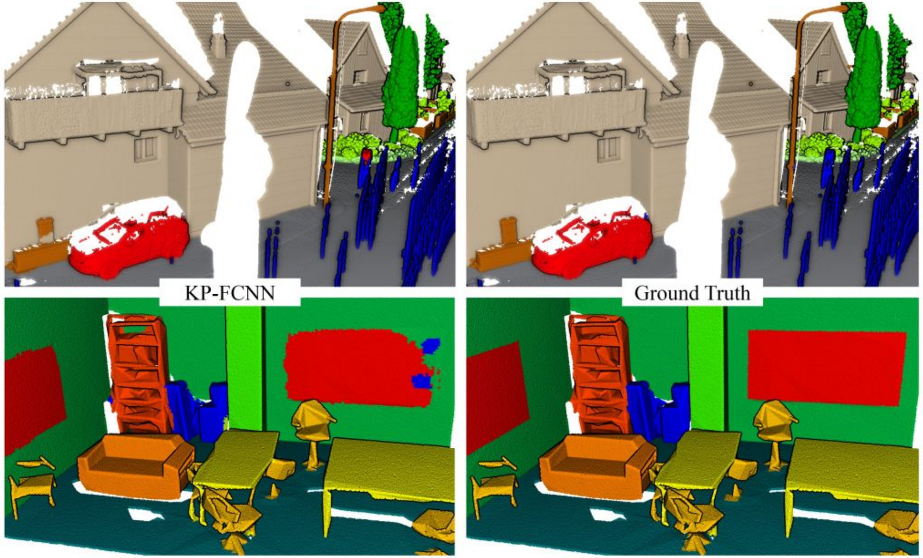


Convolution-based approaches perform better than PointNet, PointNet++, EdgeConv

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

KPConv Performance

	ModelNet40	ShapeNetPart	
Methods	OA	mcIoU	mIoU
SPLATNet [34]	-	83.7	85.4
SGPN [42]	-	82.8	85.8
3DmFV-Net [9]	91.6	81.0	84.3
SynSpecCNN [48]	-	82.0	84.7
RSNet [15]	-	81.4	84.9
SpecGCN [40]	91.5	-	85.4
PointNet++ [27]	90.7	81.9	85.1
SO-Net [19]	90.9	81.0	84.9
PCNN by Ext [2]	92.3	81.8	85.1
SpiderCNN [45]	90.5	82.4	85.3
MCConv [13]	90.9	-	85.9
FlexConv [10]	90.2	84.7	85.0
PointCNN [20]	92.2	84.6	86.1
DGCNN [43]	92.2	85.0	84.7
SubSparseCNN [9]	-	83.3	86.0
KPConv *rigid*	**92.9**	85.0	86.2
KPConv *deform*	92.7	**85.1**	**86.4**



Convolution-based approaches perform better than PointNet, PointNet++, EdgeConv

@2019

Point Clouds

PointNet

PointNet++

EdgeConv

KPConv

Point Transformer

Convolution based architectures for Point Cloud

Point Transformers

Based on the idea of attention

Attention based architectures gained popularity in NLP and Computer Vision

Attention Is All You Need

2017

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

rrent or
The best
ttention
sformer,

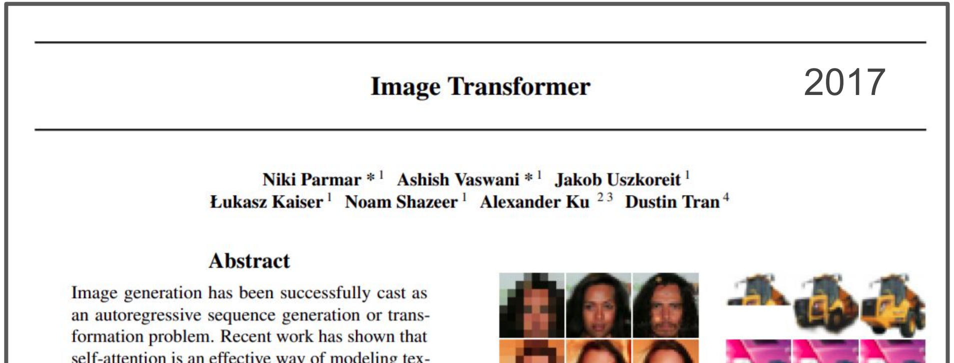


Image Transformer

2017

Niki Parmar *[1] Ashish Vaswani *[1] Jakob Uszkoreit[1]
Łukasz Kaiser[1] Noam Shazeer[1] Alexander Ku[2,3] Dustin Tran[4]

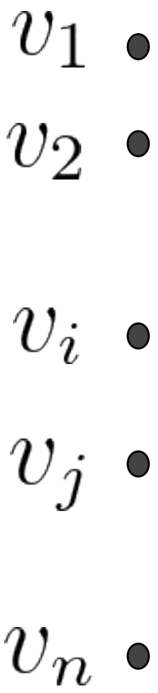
Abstract

Image generation has been successfully cast as an autoregressive sequence generation or transformation problem. Recent work has shown that self-attention is an effective way of modeling tex-

Attention

Collection of points

Attention



Each point has a value

Attention

$v_1 \bullet k_1$

$v_2 \bullet k_2$

$v_i \bullet k_i$

$v_j \bullet k_j$

$v_n \bullet k_n$

Each point has a value and a key

Attention

$v_1 \bullet k_1$

$v_2 \bullet k_2$

$v_i \bullet k_i$

$v_j \bullet k_j$

$v_n \bullet k_n$

Query q

In comes a query q

Attention

Query q

- v_1 k_1
- v_2 k_2
- v_i k_i
- v_j k_j
- v_n k_n

Output $= v_{i^*}$

$$i^* = \arg\max_i q^T k_i$$

Output value, who's key matches the query

Attention

Query q

$v_1 \bullet k_1$

$v_2 \bullet k_2$

$v_i \bullet k_i$

$v_j \bullet k_j$

$v_n \bullet k_n$

$$\text{Output} = \sum_i (q^T k_i) \cdot v_i$$

Or more like a weighted average

Attention to Point Cloud

Query q

$v_1 \bullet k_1$

$v_2 \bullet k_2$

$v_i \bullet k_i$

$v_j \bullet k_j$

$v_n \bullet k_n$

$$\text{Output} = \sum_i \left(q^T k_i\right) \cdot v_i$$

How to develop this idea for an architecture over point clouds?

Attention to Point Cloud

Query q

$\cancel{v_1} \bullet \cancel{k_1} \quad x_1$

$\cancel{v_2} \bullet \cancel{k_2} \quad x_2$

$\cancel{v_i} \bullet \cancel{k_i} \quad x_i$

$\cancel{v_j} \bullet \cancel{k_j} \quad x_j$

$\cancel{v_n} \bullet \cancel{k_n} \quad x_n$

$$\text{Output} = \sum_i (q^T k_i) \cdot v_i$$

We don’t have values and keys.

We have position, input features.

Attention to Point Cloud

Query $\cancel{q}$ x_j

$\cancel{v_1} \bullet \cancel{k_1}$ x_1

$\cancel{v_2} \bullet \cancel{k_2}$ x_2

$\cancel{v_i} \bullet \cancel{k_i}$ x_i

$\cancel{v_j} \bullet \cancel{k_j}$ x_j

$\cancel{v_n} \bullet \cancel{k_n}$ x_n

$$\text{Output} = \sum_i (q^T k_i) \cdot v_i$$

Query is a point on the point cloud

Attention to Point Cloud

$v_1 \bullet k_1$

$v_2 \bullet k_2$

$v_i \bullet k_i$

$v_j \bullet k_j$

$v_n \bullet k_n$

Query q $\quad q = \phi(x_j)$

$$\text{Output} = \sum_i (q^T k_i) \cdot v_i$$

$$v_i = \alpha(x_i)$$

$$k_i = \psi(x_i)$$

Use trainable functions (MLP) to obtain key, value, and query from features vectors x_i

Attention to Point Cloud

Query q $\qquad q = \phi(x_j)$

$v_1 \bullet k_1$

$v_2 \bullet k_2$

$v_i \bullet k_i$

$v_j \bullet k_j$

$v_n \bullet k_n$

$$x'_j = \sum_i \rho(\phi(x_j)^T \psi(x_i)) \cdot \alpha(x_i)$$

$$v_i = \alpha(x_i)$$
$$k_i = \psi(x_i)$$

Generates update for point j

Zhao et al. "Point Transformer" 2020

Point Transformer

Basic version

$$x'_j = \sum_{i \in N(x_j)} \rho(\phi(x_j)^T \psi(x_i)) \cdot \alpha(x_i)$$

Zhao et al. “Point Transformer” 2020

Point Transformer

Basic version

$$x'_j = \sum_{i \in N(x_j)} \rho(\phi(x_j)^T \psi(x_i)) \cdot \alpha(x_i)$$

Incorporating point feature + location; and using vector for attention

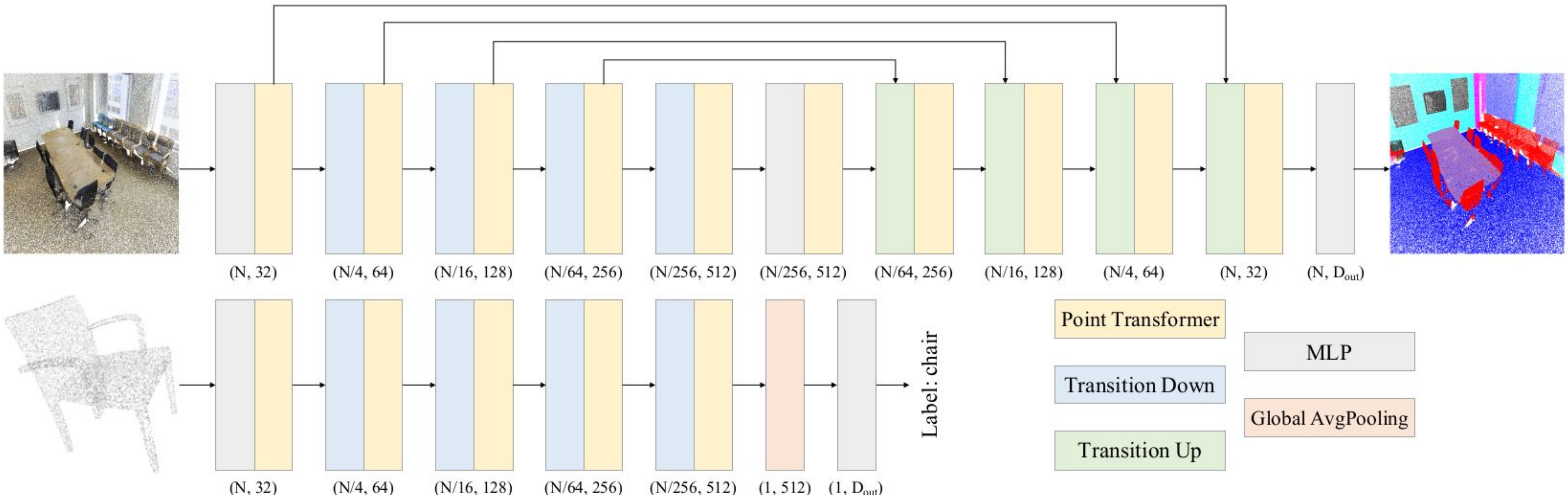
$$x'_j = \sum_{i \in N(x_j)} \rho[\beta(\phi(x_j), \psi(x_i)) + \delta(p_j - p_i)] \odot \alpha(x_i)$$

function other than dot product

position of points

Zhao et al. "Point Transformer" 2020

Point Transformer



Pooing, un-pooling, and residual connections similar to PointNet++

Zhao et al. “Point Transformer” 2020

Point Transformer

Object Classification (ModelNet40)

Method	input	mAcc	OA
3DShapeNets [43]	voxel	77.3	84.7
VoxNet [20]	voxel	83.0	85.9
Subvolume [23]	voxel	86.0	89.2
MVCNN [30]	image	–	90.1
PointNet [22]	point	86.2	89.2
PointNet++ [24]	point	–	91.9
SpecGCN [36]	point	–	92.1
PointCNN [18]	point	88.1	92.2
DGCNN [40]	point	90.2	92.2
PointWeb [50]	point	89.4	92.3
SpiderCNN [44]	point	–	92.4
PointConv [42]	point	–	92.5
KPConv [33]	point	–	92.9
InterpCNN [19]	point	–	93.0
PointTransformer	point	**90.6**	**93.7**

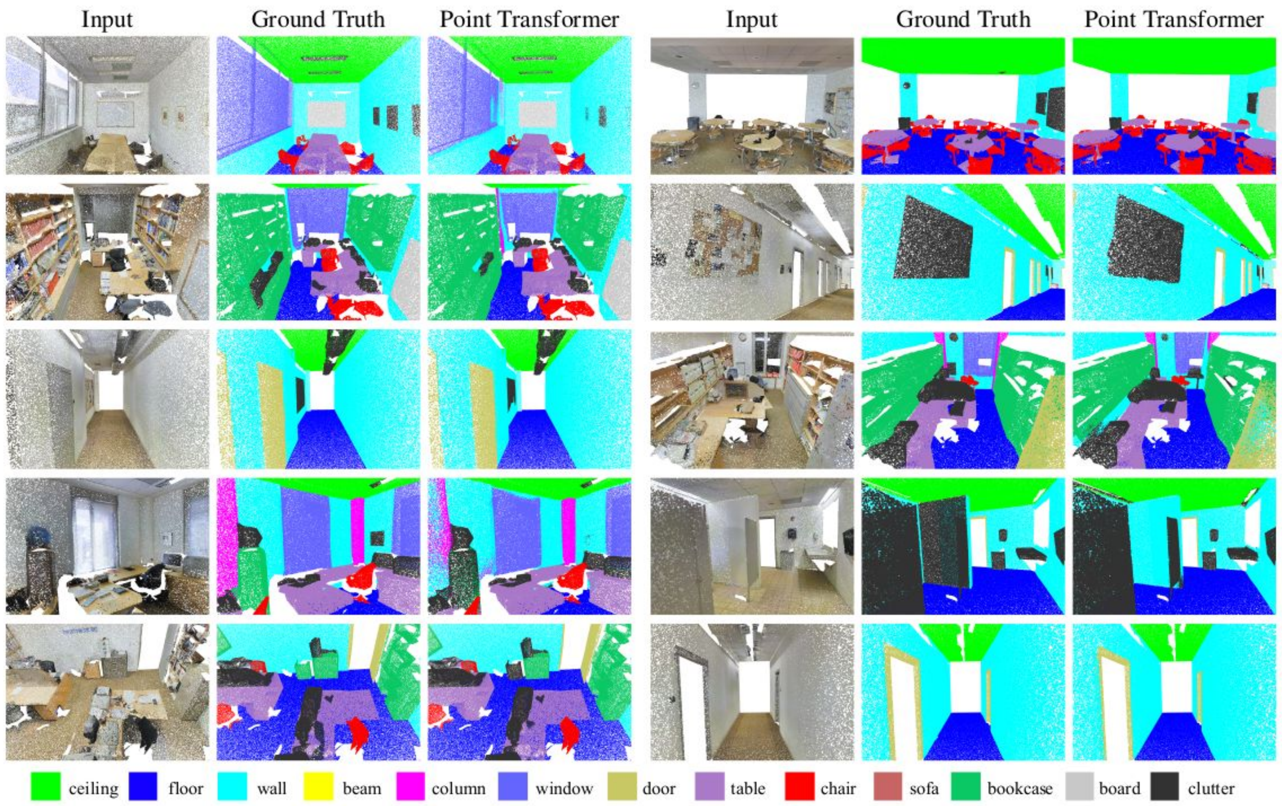
Object Part Segmentation
(ShapeNetPart Dataset)

Method	cat. mIoU	ins. mIoU
PointNet [22]	80.4	83.7
PointNet++ [24]	81.9	85.1
SPLATNet	83.7	85.4
SpiderCNN [44]	81.7	85.3
PCNN [38]	81.8	85.1
PointCNN [18]	84.6	86.1
DGCNN [40]	82.3	85.1
SGPN [39]	82.8	85.8
PointConv [42]	82.8	85.7
InterpCNN [19]	84.0	86.3
KPConv [33]	**85.1**	86.4
PointTransformer	83.7	**86.6**

State-of-the-art @2020

Zhao et al. “Point Transformer” 2020

Point Transformer



Semantic Segmentation on S3DIS Dataset

https://paperswithcode.com/sota/semantic-segmentation-on-s3dis

State-of-the-art @2020

Zhao et al. "Point Transformer" 2020

Point Cloud-based Architectures

Efficient than voxel based architectures

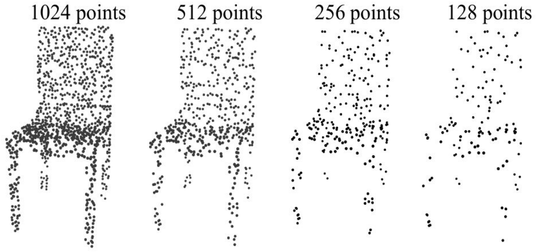
Suitable for point cloud inputs (LiDAR or RGB-D)

Point Cloud-based Architectures

Efficient than voxel based architectures

Suitable for point cloud inputs (LiDAR or RGB-D)

Point clouds may not be the best way to represent 3D shapes

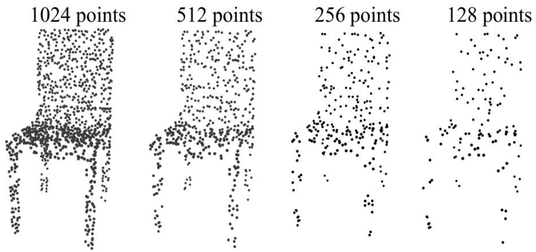


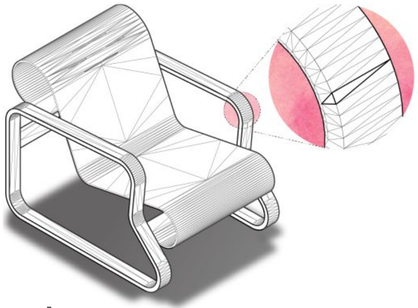
Point Cloud-based Architectures

Efficient than voxel based architectures

Suitable for point cloud inputs (LiDAR or RGB-D)

Point clouds may not be the best way to represent 3D shapes



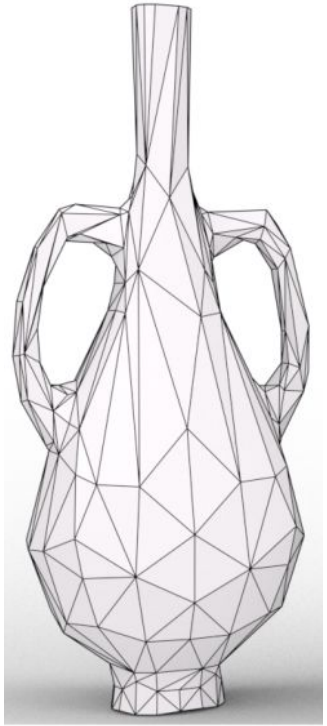


Mesh

Mesh

Mesh Representation

Mesh = Vertices, Faces, Edges



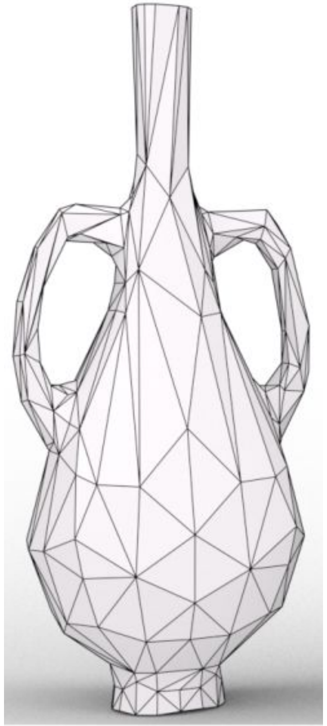
source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

Mesh Representation

Mesh = Vertices, Faces, Edges

3d locations

$$v = (x, y, z)$$



source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

Mesh Representation

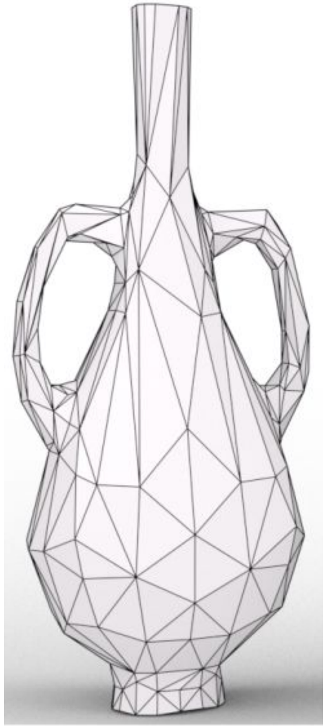
Mesh = Vertices, Faces, Edges

3d locations

$$v = (x, y, z)$$

Triplet of vertices

$$f = (v_1, v_2, v_3)$$



source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

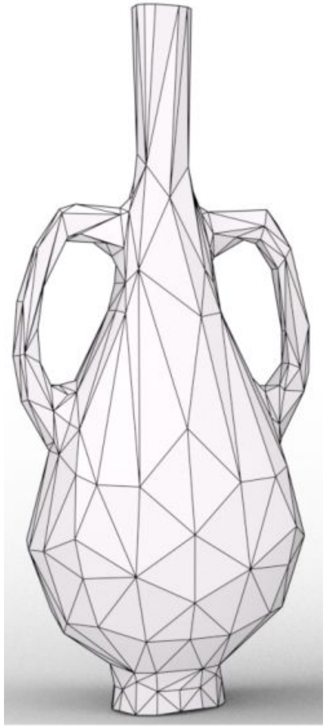
Mesh Representation

Mesh = Vertices, Faces, Edges

3d locations
$v = (x, y, z)$

Triplet of vertices
$$f = (v_1, v_2, v_3)$$

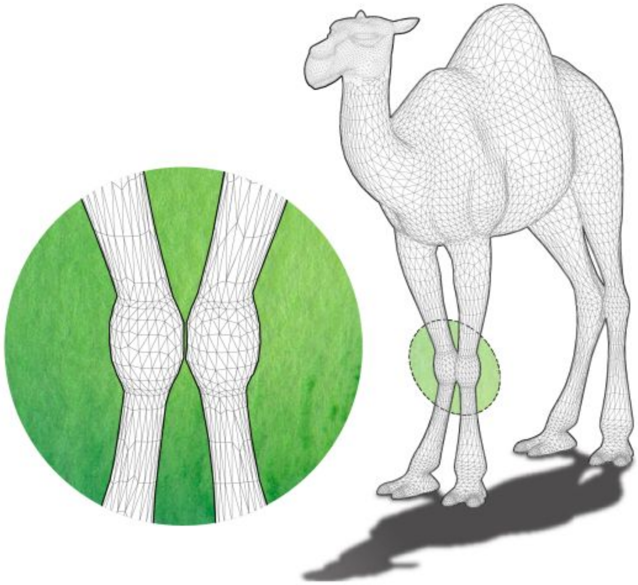
Pair of vertices
$$e = (v_1, v_2)$$



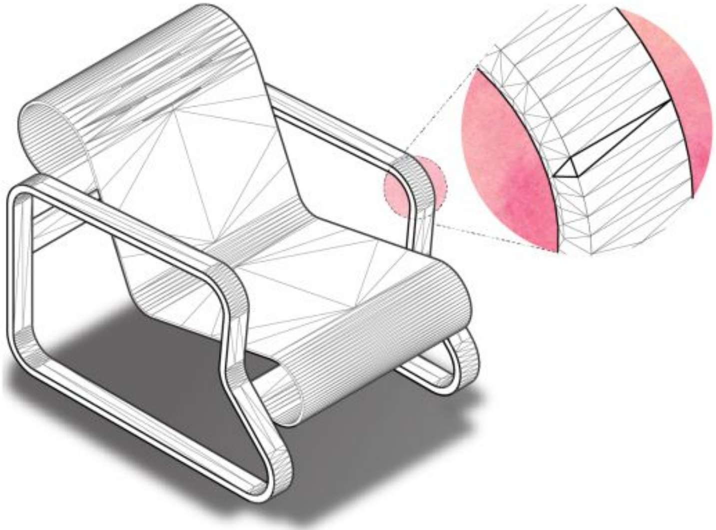
source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

Mesh Representation

Conveys distinctness of local shape



Adaptive to non-uniform shape



source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

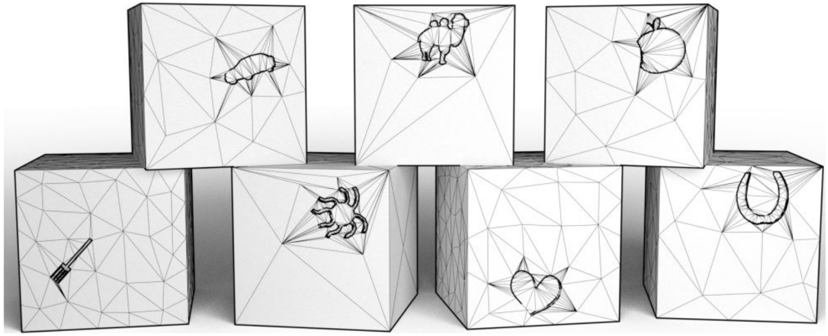
Learning on Meshes

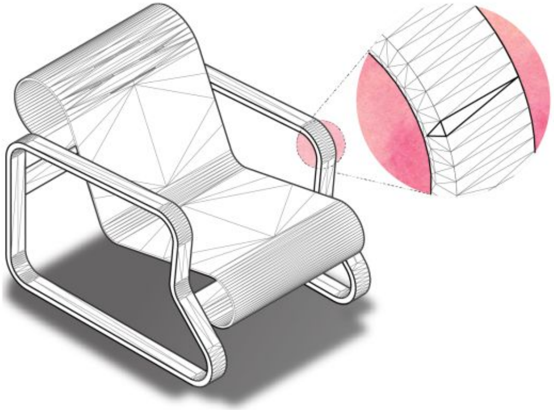
Architectures should be able to exploit this property

Learning on Meshes

Architectures should be able to exploit this property

Problem: non-uniformity of the mesh



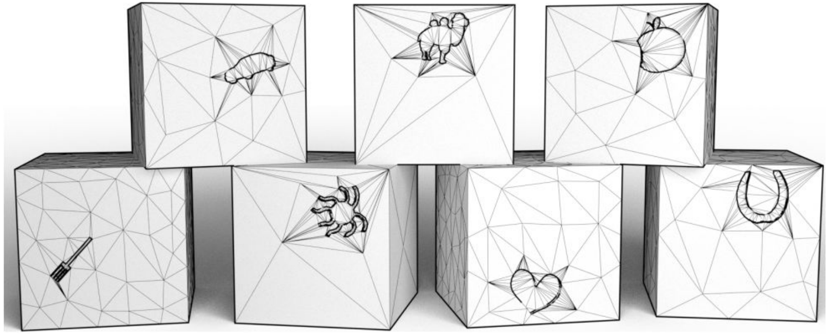


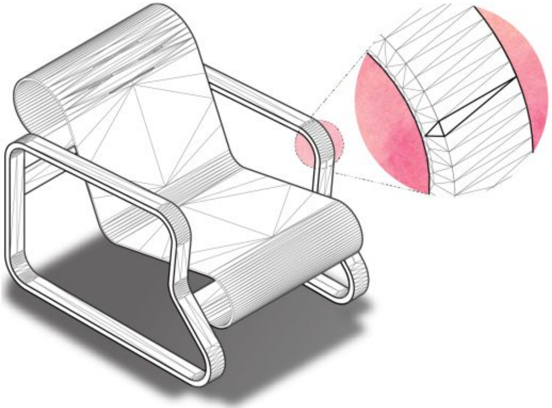
source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

Learning on Meshes

How do we define convolution, pooling, and unpooling on this?

Problem: non-uniformity of the mesh





source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

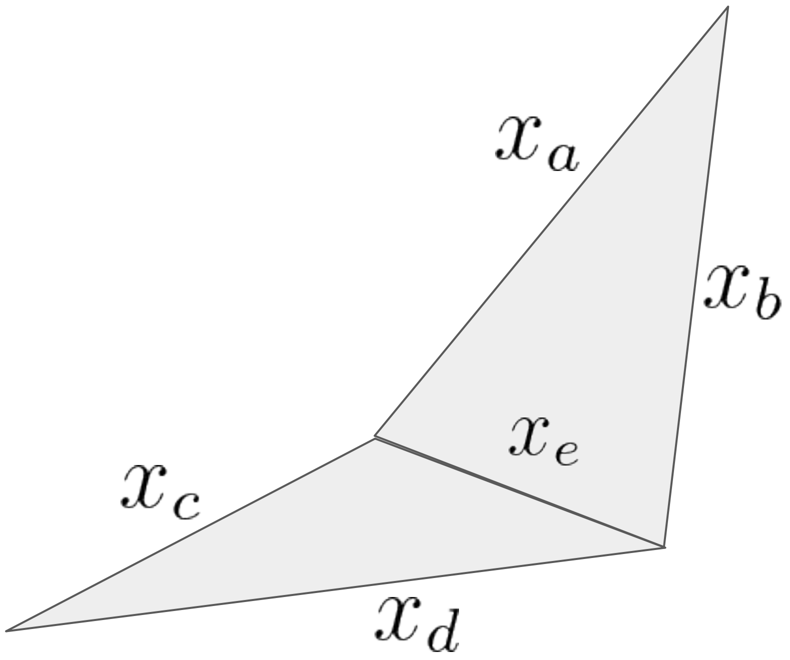
MeshCNN

Operates over mesh edges

MeshCNN

Operates over mesh edges

Generates and updates representation vectors over mesh edges

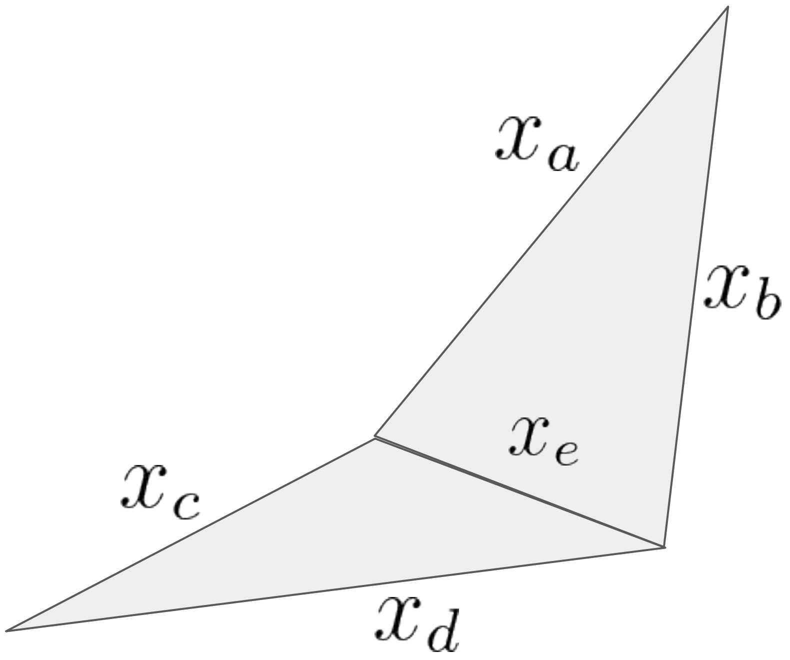


MeshCNN

Operates over mesh edges

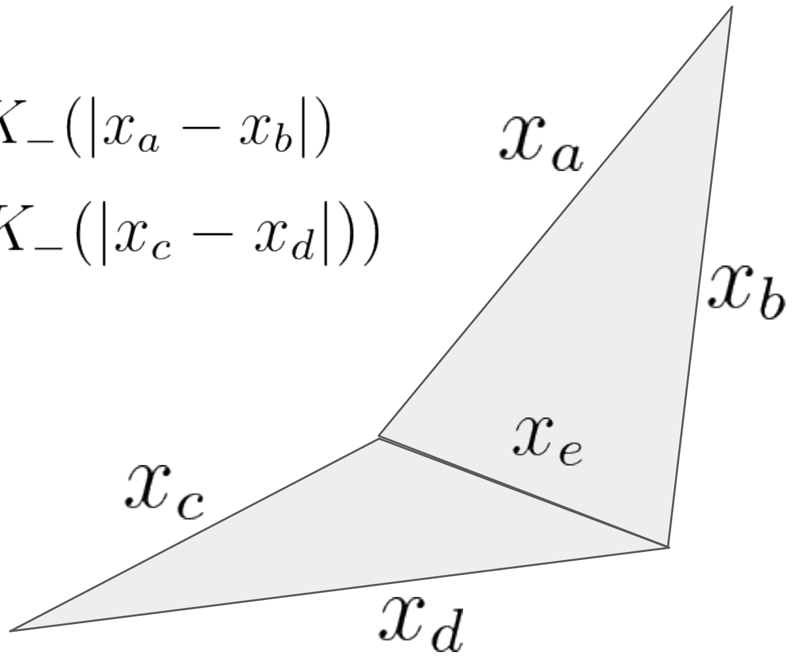
Generates and updates representation vectors over mesh edges

for manifold mesh every edge has two adjacent faces and four adjacent edges



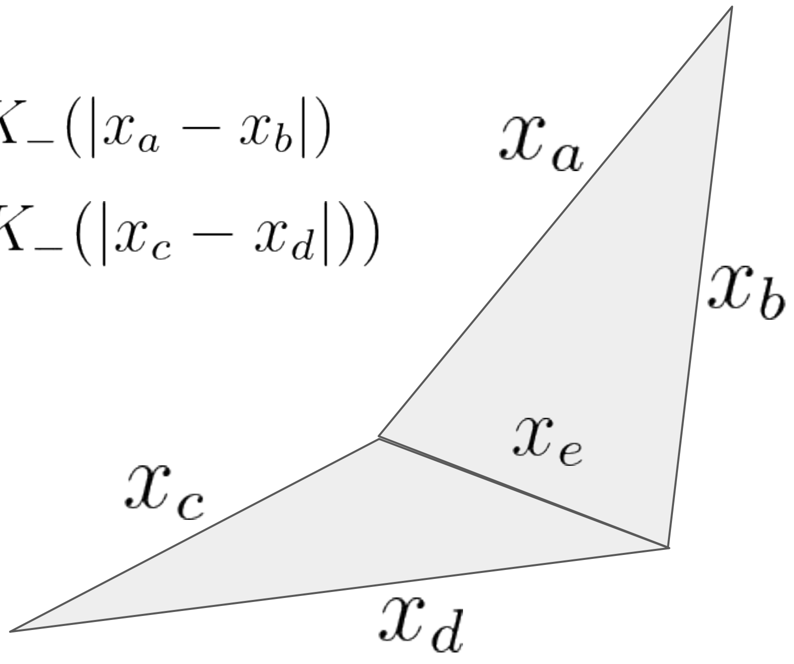
Updating Edge Features

$$x'_e = \sigma(Kx_e + K_+(x_a + x_b) + K_-(|x_a - x_b|)$$
$$+K_+(x_c + x_d) + K_-(|x_c - x_d|))$$



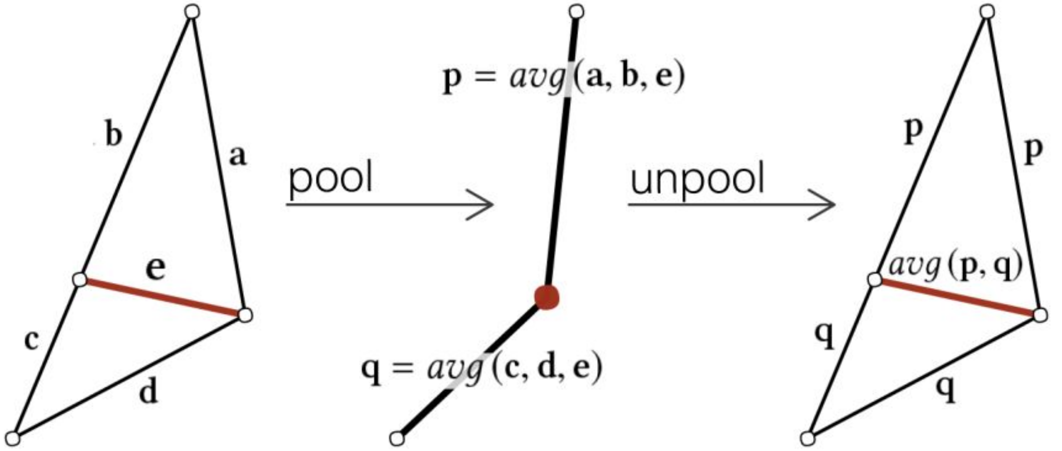
Updating Edge Features

$$x'_e = \sigma(Kx_e + K_+(x_a + x_b) + K_-(|x_a - x_b|)$$
$$+K_+(x_c + x_d) + K_-(|x_c - x_d|))$$



Invariant to ordering of neighboring edges

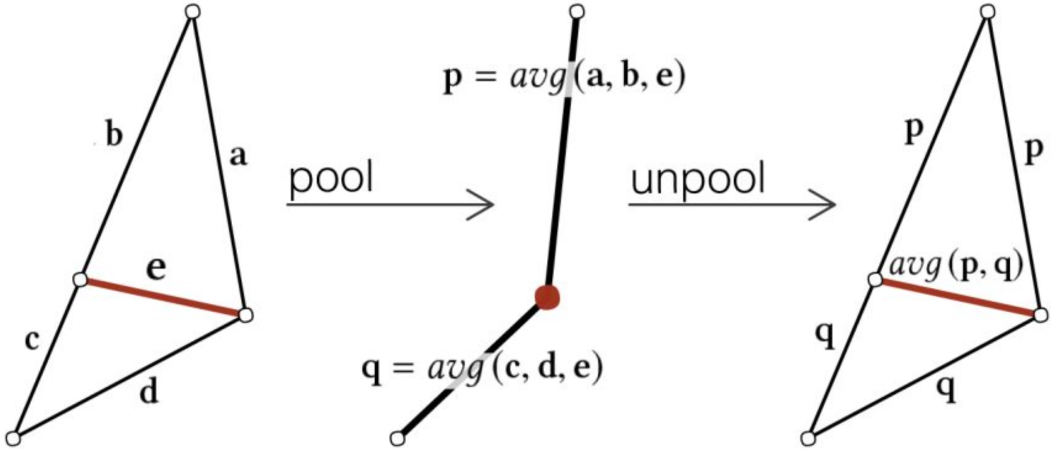
Pooling and Unpooling



In the figure a is h_a ...

source: Hanocka et al. "MeshCNN: A Network with an Edge" ACM Trans. Graph. 2019

Pooling and Unpooling

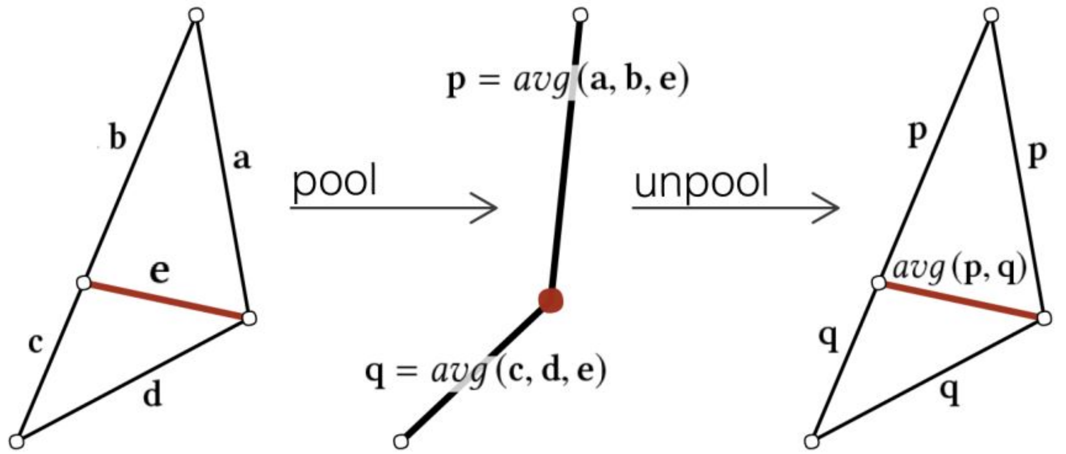


edges with N largest feature vector are collapsed at each layer

In the figure a is h_a ...

source: Hanocka et al. "MeshCNN: A Network with an Edge" ACM Trans. Graph. 2019

Pooling and Unpooling



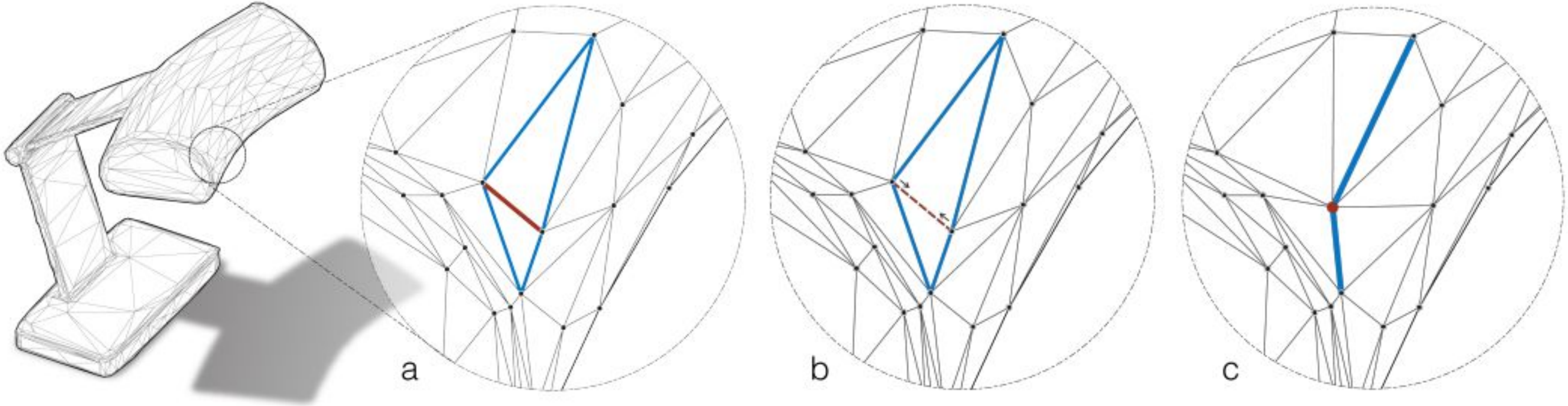
edges with N largest feature vector are collapsed at each layer

in L2 norm $||e||_2$

In the figure a is h_a ...

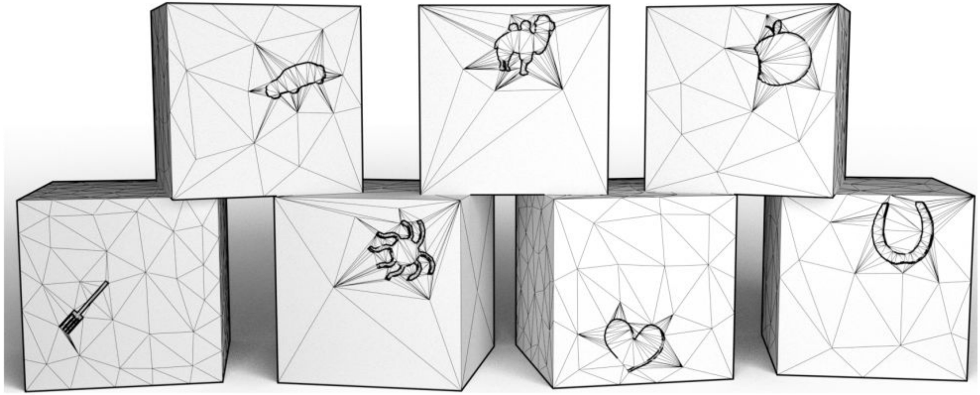
source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

Pooling and Unpooling



source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

MeshCNN: Interesting Results



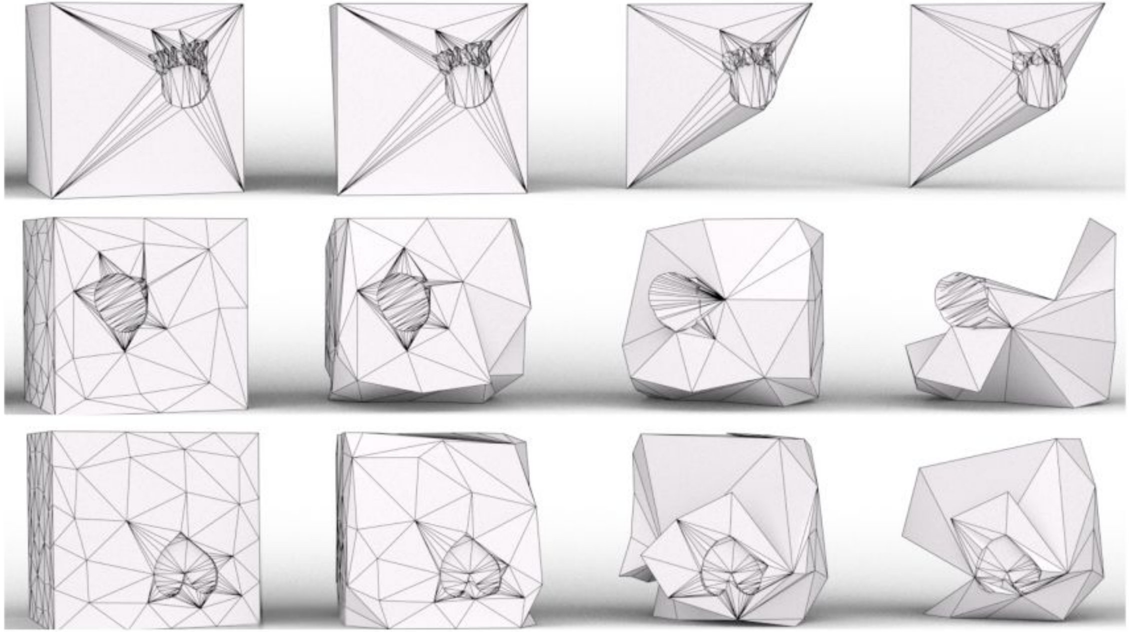
Classifying fine engraved cubes

Cube Engraving Classification

method	input res	test acc
MeshCNN	750	**92.16**%
PointNet++	4096	64.26%

source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

MeshCNN: Interesting Results

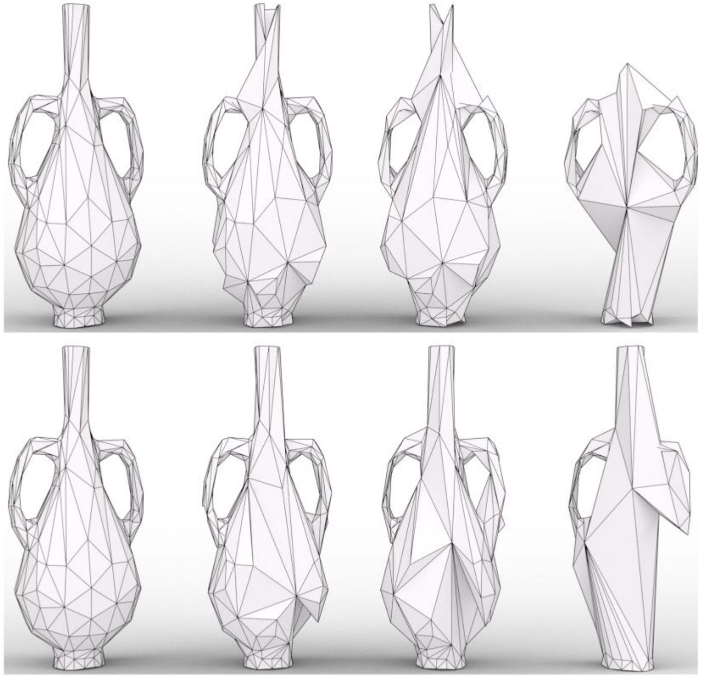


preserves important edges required for the task

depth

source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

MeshCNN: Interesting Results



depth

Task 1: Vaze has a handle?

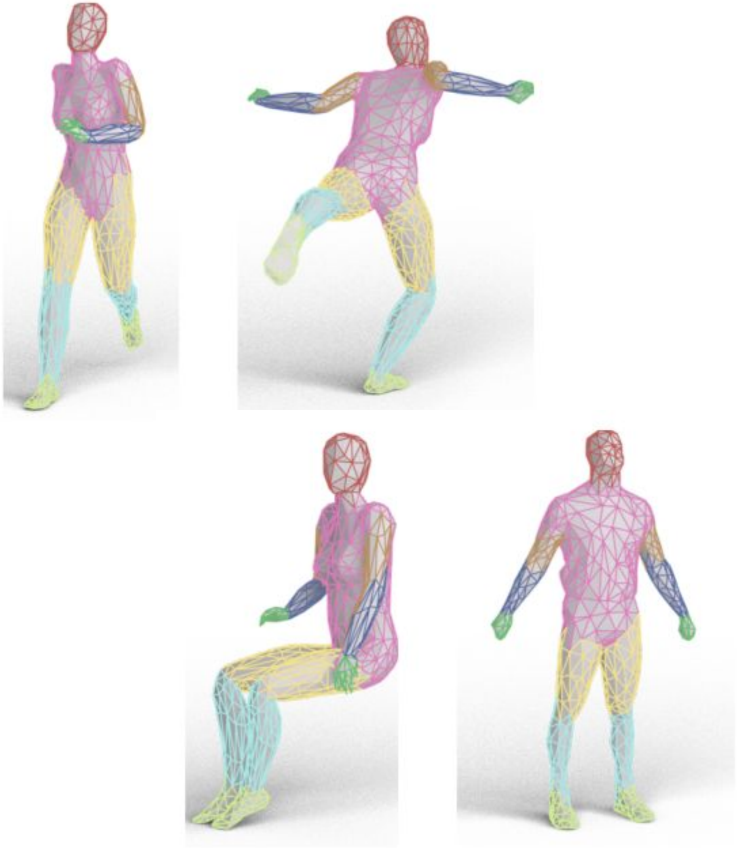
Task 2: Vaze has a neck?

source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

MeshCNN: Human Shape Segmentation

Human Body Segmentation		
Method	# Features	Accuracy
MeshCNN	5	**92.30%**
SNGC	3	91.02%
Toric Cover	26	88.00%
PointNet++	3	90.77%
DynGraphCNN	3	89.72%
GCNN	64	86.40%
MDGCNN	64	89.47%

[2018]



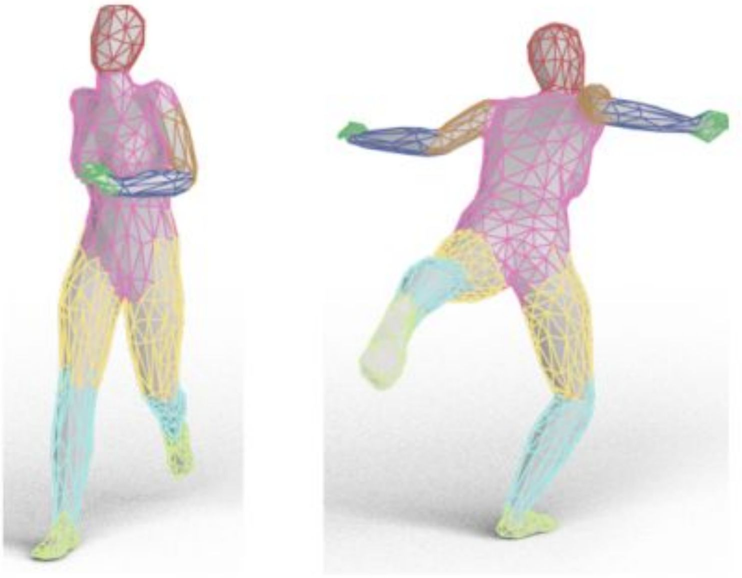
source: Hanocka et al. "MeshCNN: A Network with an Edge" ACM Trans. Graph. 2019

Mesh based Architectures

More structure. Opportunity for the architecture to be more expressive.

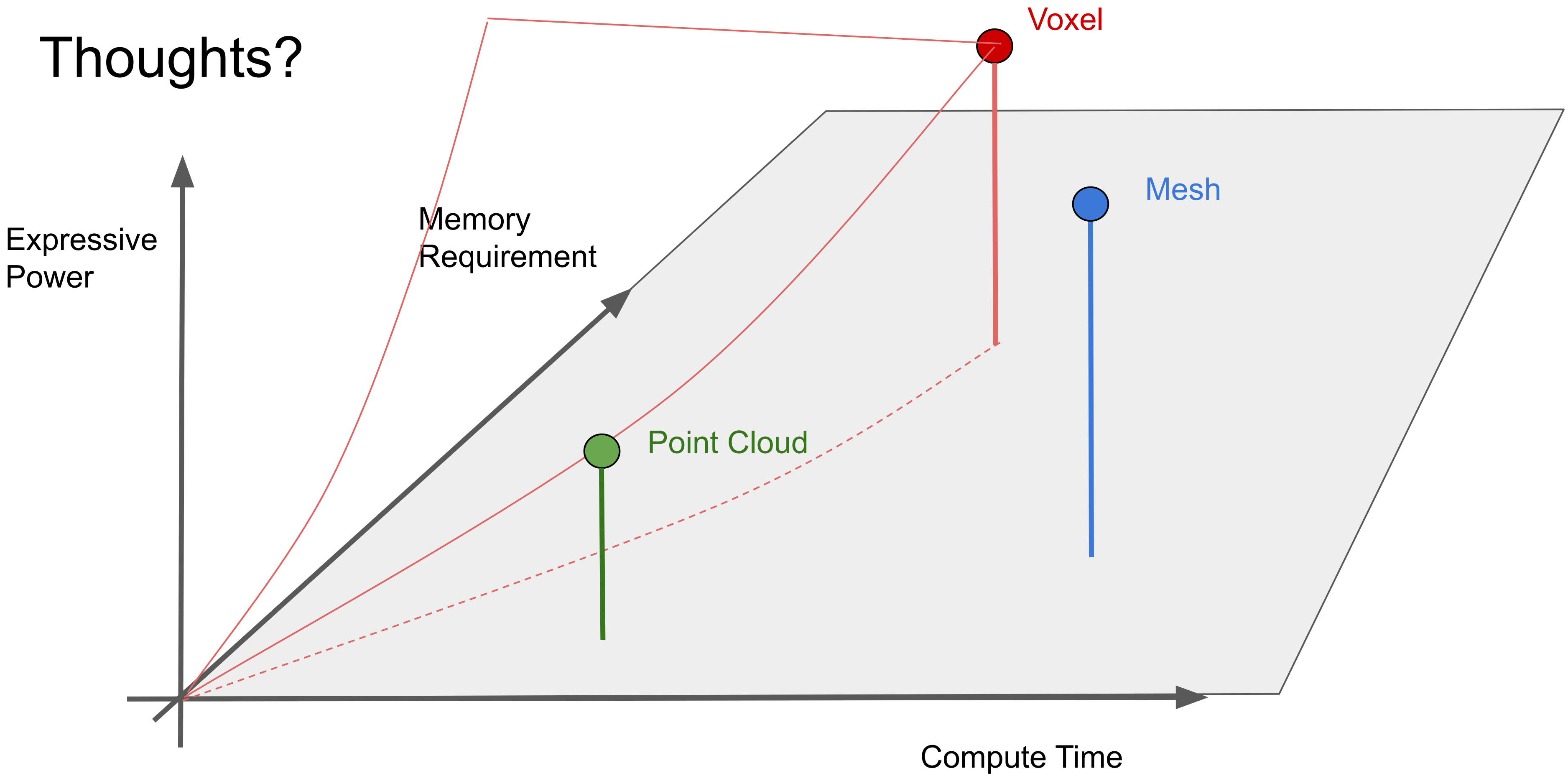
Computationally expensive than Point Cloud based architectures.

- Pooling, unpooling, manifoldness



source: Hanocka et al. “MeshCNN: A Network with an Edge” ACM Trans. Graph. 2019

Thoughts?


Voxel
Mesh
Expressive Power
Memory Requirement
Point Cloud
Compute Time

Conclusion

- Need for semantic understanding
- Need for Deep Learning Models on richer domains
  - Voxels, Point Clouds, Meshes, Graphs …
- Deep Learning architectures for 3D
  - Voxel
  - Point Cloud
  - Mesh
- Dataset and Software

Conclusion

- Need for semantic understanding
- Need for Deep Learning Models on richer domains
  - Voxels, Point Clouds, Meshes, Graphs …
- Deep Learning architectures for 3D
  - Voxel
  - Point Cloud
  - Mesh
- Dataset and Software

Next: A unifying view for constructing DL models

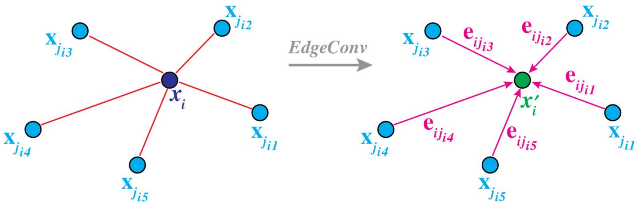
Backup

Conclusion: Architectures Discussed

Voxel

VoxNet

OctNet



Point Cloud

PointNet

EdgeConv

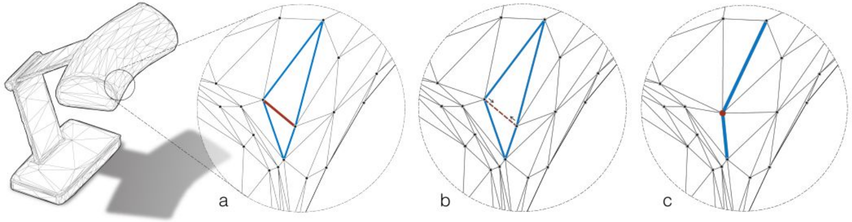
Point Transformer

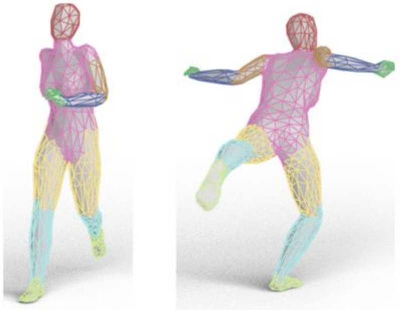
PointNet++

KPConv

Mesh

MeshCNN





Software

PyTorch Geometric

https://www.pytorch-geometric.readthedocs.io/

Open 3D

http://www.open3d.org/

Datasets

Object Classification and Object Part Segmentation

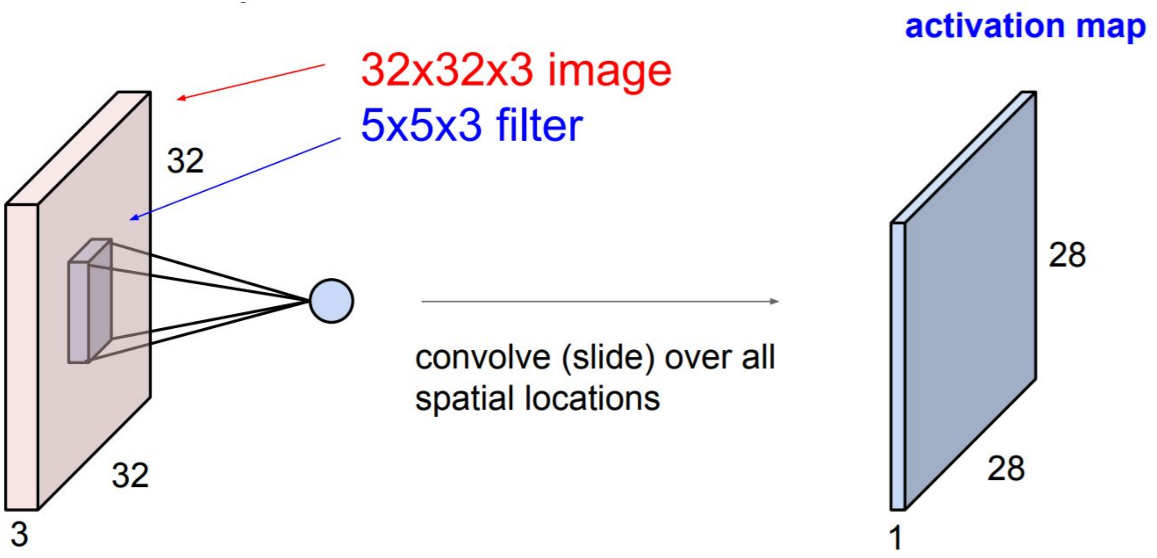
- ModelNet
- ShapeNet

3D Scene Segmentation

- ScanNet
- Stanford 3D Indoor Scene Dataset (S3DIS)
- Semantic KITTI
- Matterport 3D

Components: Convolution, Pooling, Un-pooling, and MLP

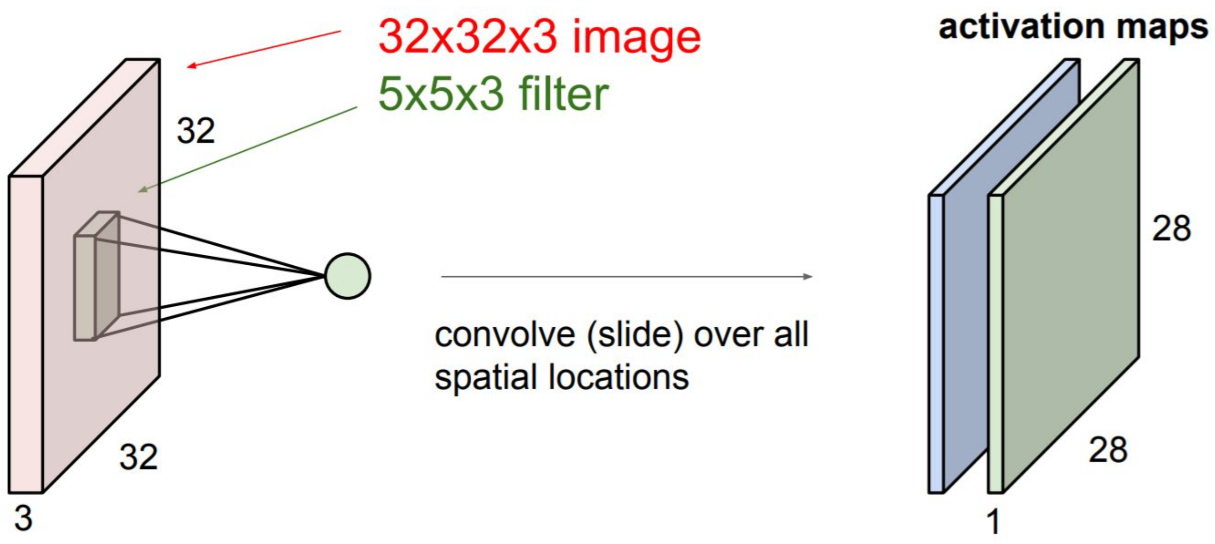
Convolution Layer



$$(f * g)(x) = \int_{\mathcal{X}} f(z)g(z-x)dz$$

Source: Fei-Fei Li, Rajat Krishna, Danfei Xu "Stanford CS231n: Convolutional Neural Networks for Visual Recognition" Spring 2021

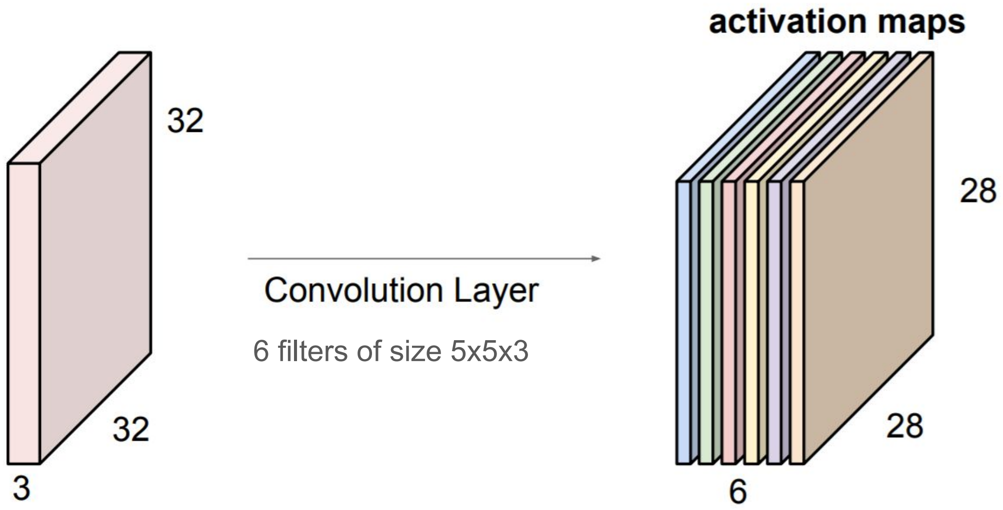
Convolution Layer



$$(f * g)(x) = \int_{\mathcal{X}} f(z)g(z-x)dz$$

Source: Fei-Fei Li, Rajat Krishna, Danfei Xu "Stanford CS231n: Convolutional Neural Networks for Visual Recognition" Spring 2021

Convolution Layer

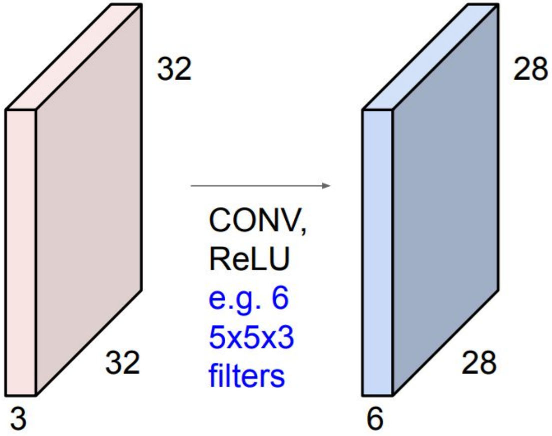


New "image" 28x28x6

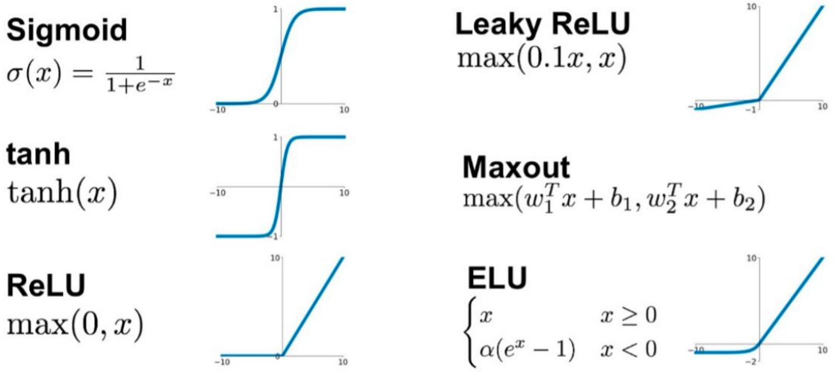
$$(f * g)(x) = \int_{\mathcal{X}} f(z)g(z-x)dz$$

Source: Fei-Fei Li, Rajat Krishna, Danfei Xu "Stanford CS231n: Convolutional Neural Networks for Visual Recognition" Spring 2021

Convolutional Neural Networks

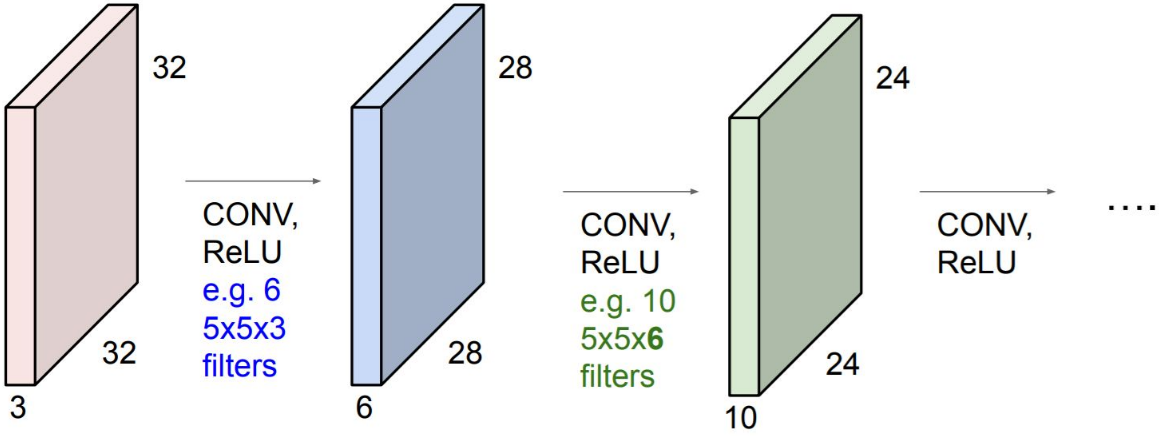


Non-linearity



Source: Fei-Fei Li, Rajat Krishna, Danfei Xu “Stanford CS231n: Convolutional Neural Networks for Visual Recognition” Spring 2021

Convolutional Neural Networks



Source: Fei-Fei Li, Rajat Krishna, Danfei Xu “Stanford CS231n: Convolutional Neural Networks for Visual Recognition” Spring 2021

Pooling Layer

MAX POOLING

2	1	3	9
3	4	4	8
6	3	4	2
1	0	2	2

→

4	9
6	4

MEAN POOLING

2	1	3	9
3	4	4	8
6	3	4	2
1	0	4	2

→

2.5	6
2.5	2.5

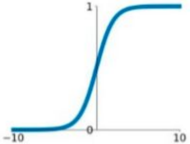
Source: Fei-Fei Li, Rajat Krishna, Danfei Xu “Stanford CS231n: Convolutional Neural Networks for Visual Recognition” Spring 2021

Fully Connected Layer

$$y = \sigma(Wx + c)$$

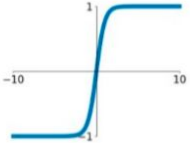
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



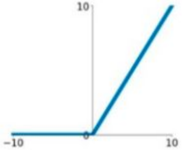
tanh

$\tanh(x)$



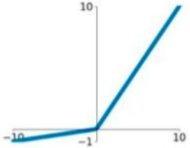
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

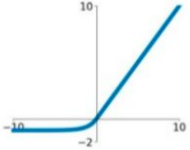


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

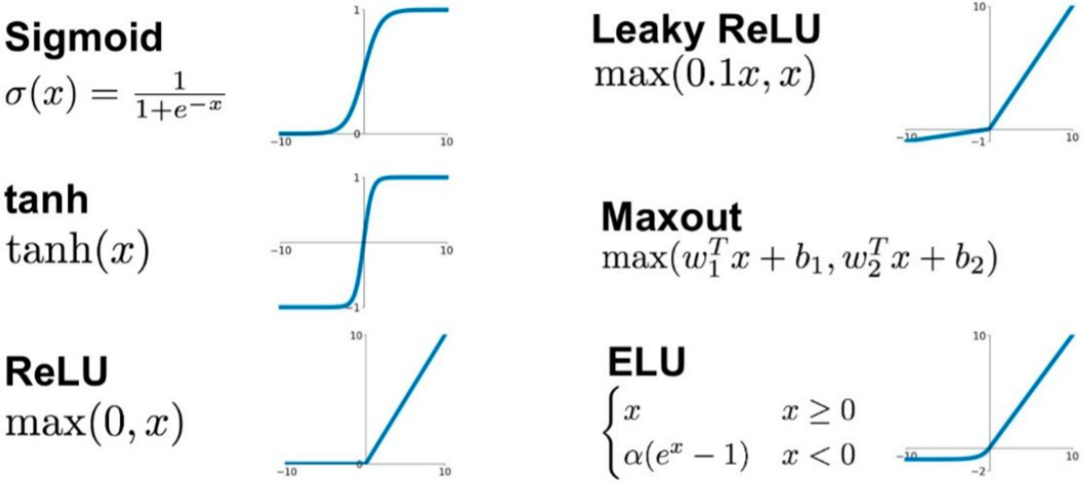
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Source: Fei-Fei Li, Rajat Krishna, Danfei Xu “Stanford CS231n: Convolutional Neural Networks for Visual Recognition” Spring 2021

Multi-Layer Perceptron (MLP)

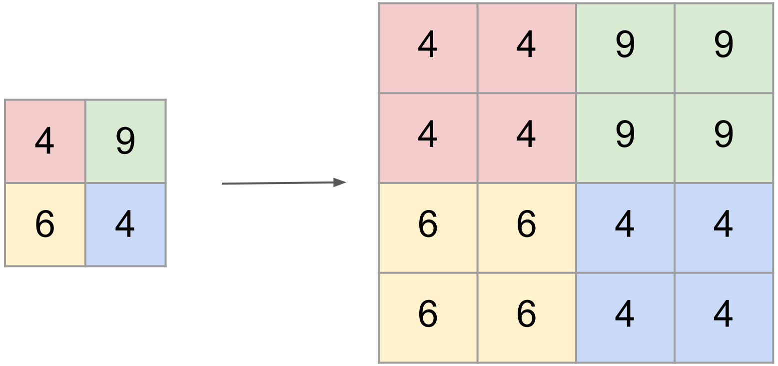
$$y = \sigma(W_t \sigma(W_{t-1} \sigma(\cdots \sigma(W_1 x + c_1) \cdots) + c_{t-1}) + c_t)$$



Source: Fei-Fei Li, Rajat Krishna, Danfei Xu “Stanford CS231n: Convolutional Neural Networks for Visual Recognition” Spring 2021

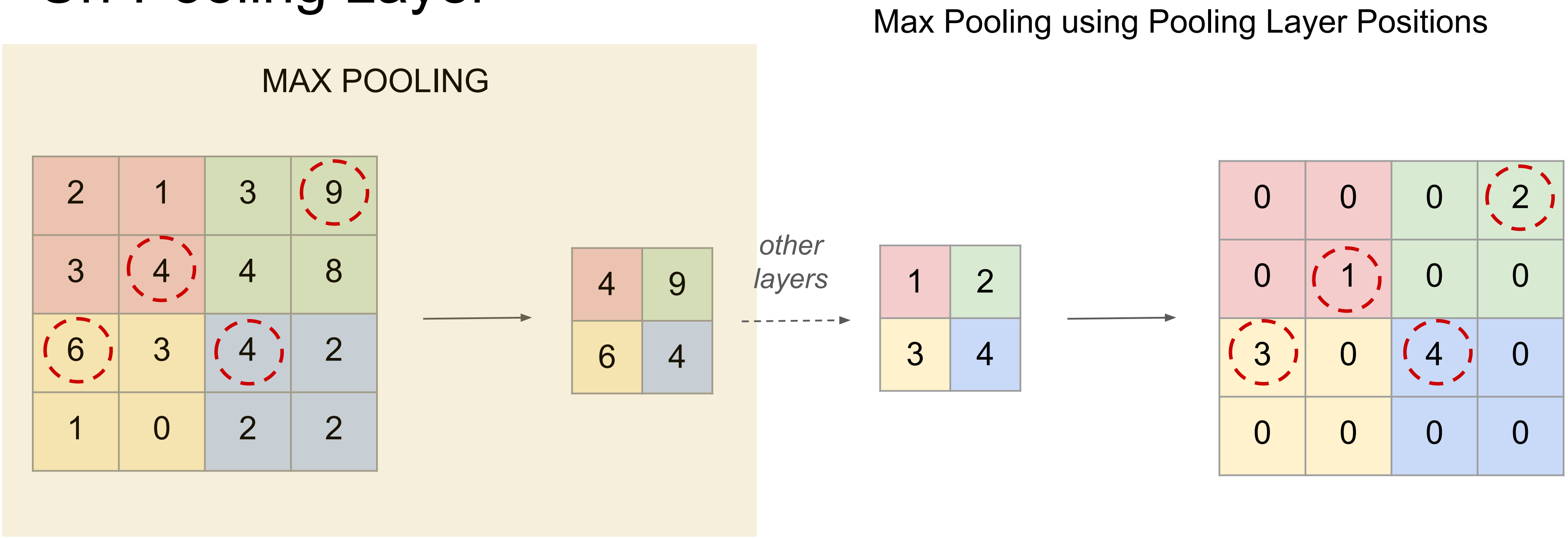
Un-Pooling Layer

Nearest Neighbor



Source: Fei-Fei Li, Rajat Krishna, Danfei Xu “Stanford CS231n: Convolutional Neural Networks for Visual Recognition” Spring 2021

Un-Pooling Layer



Source: Fei-Fei Li, Rajat Krishna, Danfei Xu "Stanford CS231n: Convolutional Neural Networks for Visual Recognition" Spring 2021