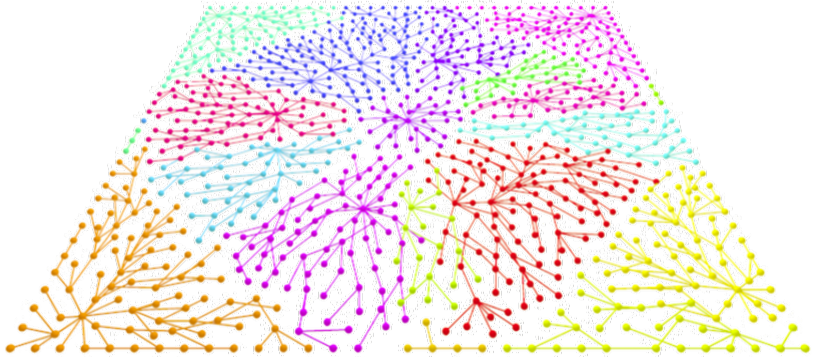
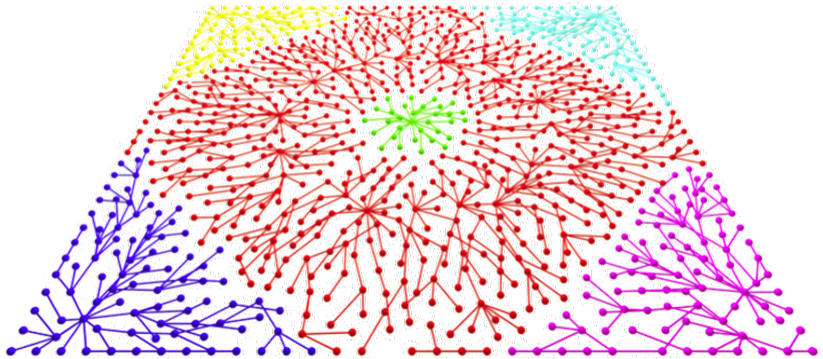
# CS233, CME251: Geometric and Topological Data Analysis

Leonidas Guibas
Computer Science Department
Stanford University





Lecture 14
17 May 2021

# Summer Geometry Institute 2021

Home | About Geometry Processing | 2021 Fellows | How to Apply | Activities and Schedule | Organization and Sponsors | Contact

## SGI 2021

The Summer Geometry Institute (SGI) is a six-week paid summer research program introducing undergraduate and graduate students to the field of **geometry processing**. Geometry processing has a long history of breakthrough developments that have guided design of 3D tools for computer vision, additive manufacturing, scientific computing, and other disciplines. Algorithms for geometry processing combine ideas from disciplines including differential geometry, topology, physical simulation, statistics, and optimization.

In the first week of SGI, participants will attend hands-on tutorials introducing the theory and practice of geometry processing; no background or previous experience is necessary. During the remaining weeks, participants will work in teams on research projects led by faculty and research scientists in this discipline, while attending talks and other sessions led by visiting researchers.

SGI will be held remotely in 2021, but participants are expected to be engaged full-time. No prior research experience or coursework in geometry processing is necessary to participate in SGI; students who have excelled in the math, science, and/or computing programs available to them are strongly encouraged to apply.

Massachusetts Institute of Technology
Cambridge MA 02139-4307
Accessibility

Login using Touchstone

https://geometryinstitute.mit.edu/

# Second Graduate Student Conference:

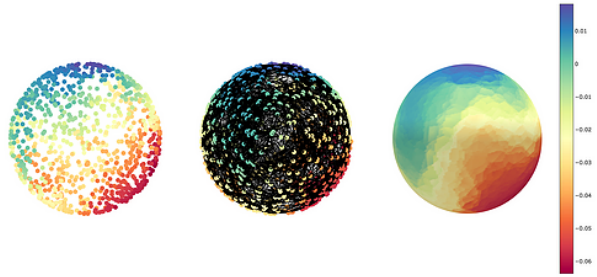
## *Geometry and Topology meet Data Analysis and Machine Learning*

*(GTDAML2021)*

*July 30 - August 1, 2021*

Contact: gtdaml2021@gmail.com

*Home*
*Program*
*Organizing Committee*
*Registration*



Welcome! We are pleased to announce that the **Second Graduate Student Conference: Geometry and Topology meet Data Analysis and Machine Learning (GTDAML2021)** will be held online from July 30 to August 1, 2021.

The goal of the conference is to bring together graduate students to share their work, interests, and presence in the flourishing research landscape connecting applications of Geometry and Topology to Data Analysis and Machine Learning. We aim to enhance discussion and collaboration via poster sessions, short presentations, and discussion panels. This is the second edition of the graduate student conference that was first held at OSU in 2019 (https://tgda.osu.edu/gtdaml2019/).

## Program

Students may apply to give a 20 minute talk (through Zoom) or a poster presentation (through gather.town). Talks and posters do not have to be about the participants' own research, and expository talks are also very welcome. Students are encouraged to apply to give a talk, but if a talk cannot be scheduled due to time limitations, students are invited to present a poster instead. We are expecting to schedule around 20 talks in total.

The program will include a special lecture by **Professor Deanna Needell** from the Department of Mathematics at UCLA.

In addition, we will have a discussion panel on industry and research. Some of our confirmed panelists include:
**Professor Lorin Crawford** (Microsoft Research and Brown University), **Professor Marco Cuturi** (Google Brain and CREST - ENSAE, Institut Polytechnique de Paris), and **Jesse Zhang** (PhD Stanford, Co-Founder at Beacons).

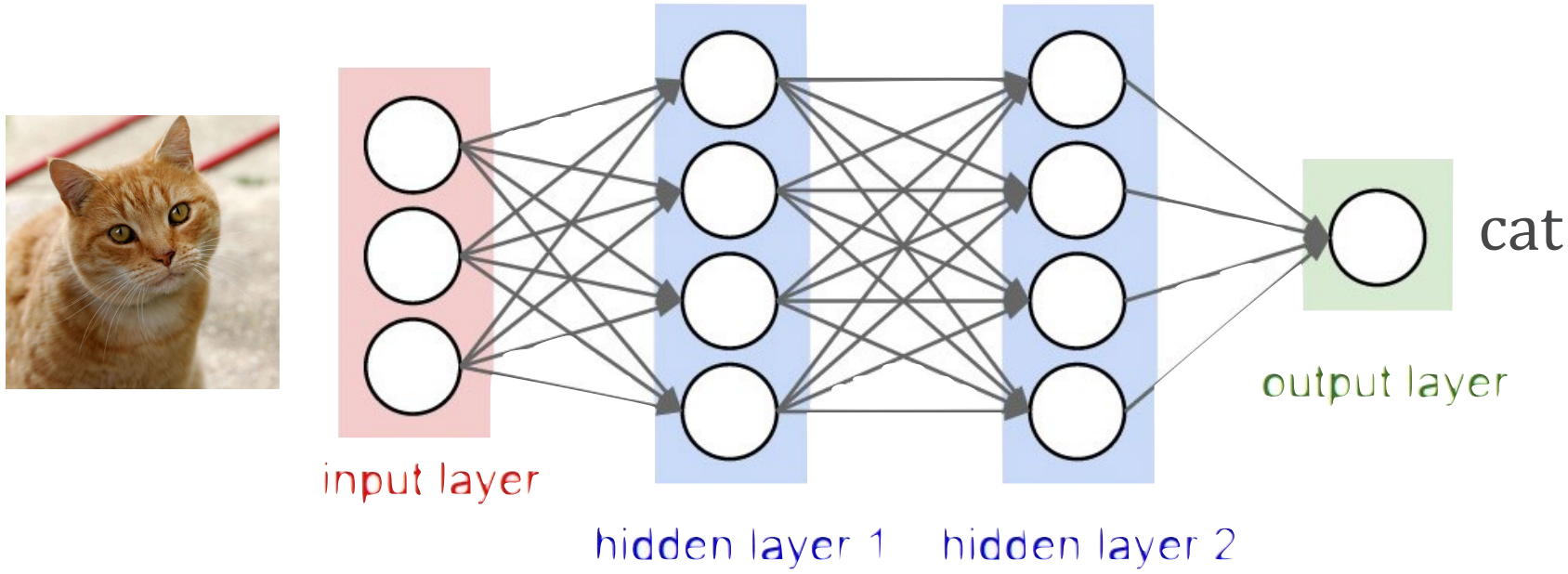
https://gtdaml.wixsite.com/2021

# Last Time: Deep Nets, Multi-View and Volumetric Approaches to 3D

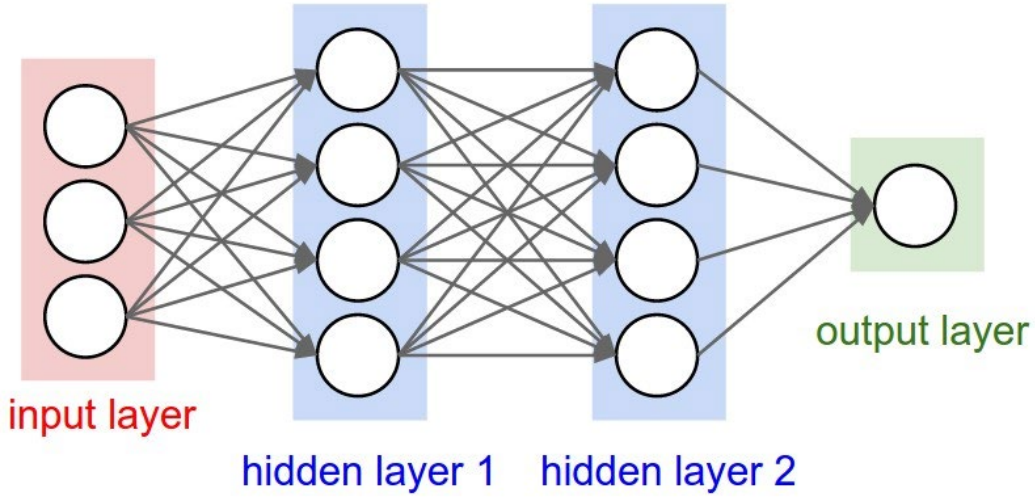
# Deep Learning

- Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.

*Deep Learning by Y. LeCun et al. Nature 2015*

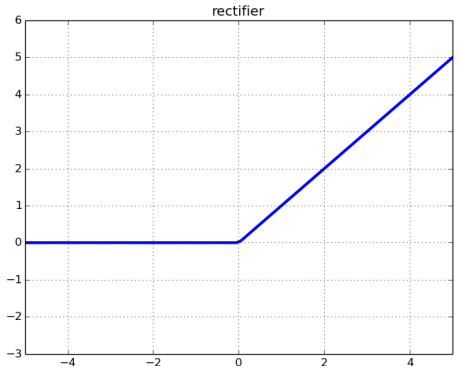


# Neural Networks Non-Linearities




$f$: non-linear activation function

$$y = \mathrm{ReLU}(x) = \max(\mathbf{0}, x)$$

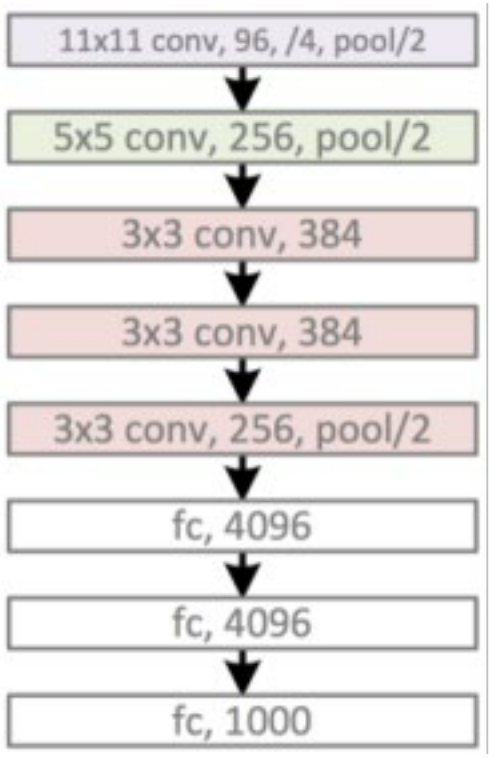


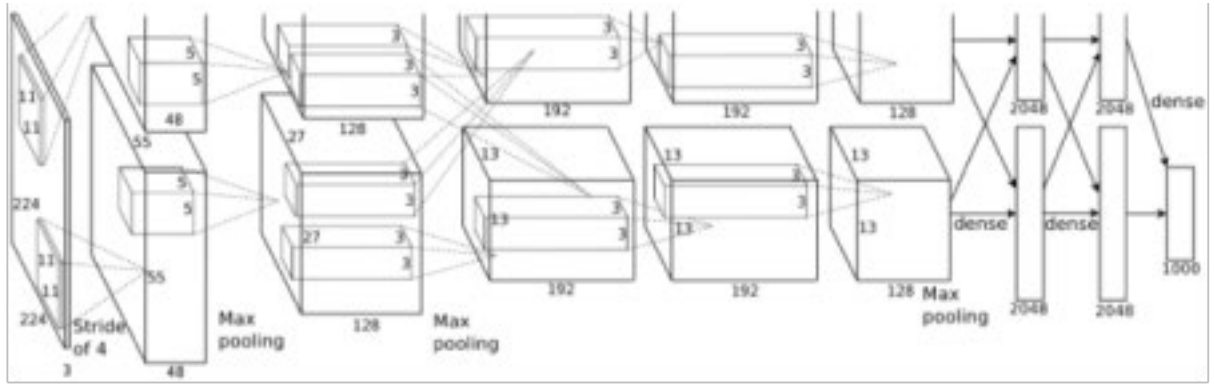

**Model:** Multi-Layer Perceptron (MLP)

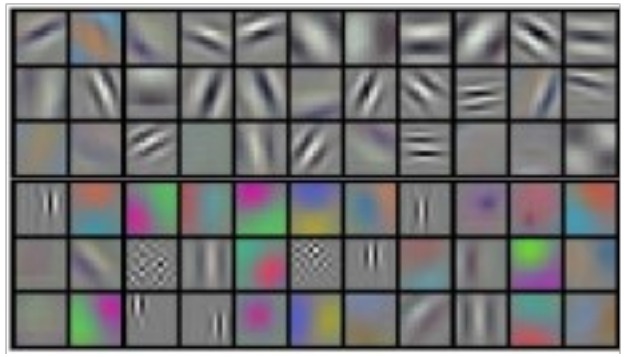
$$y' = W_3 f(W_2 f(W_1 x + b_1) + b_2) + b_3)$$

# Convolutional Neural Networks

AlexNet

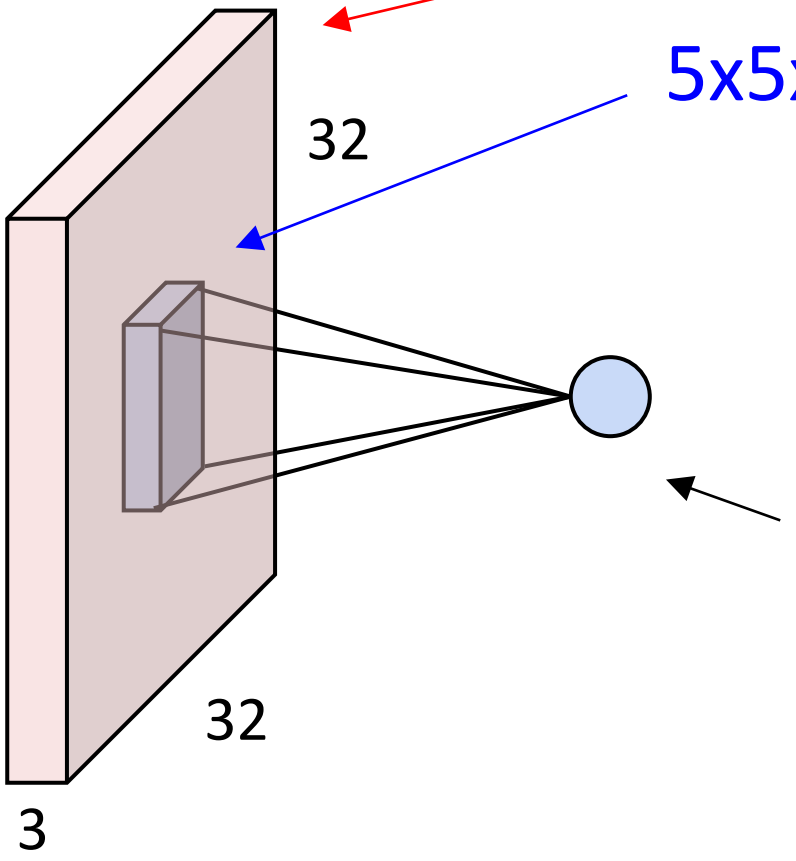


The first work that popularized Convolutional Networks in Computer Vision

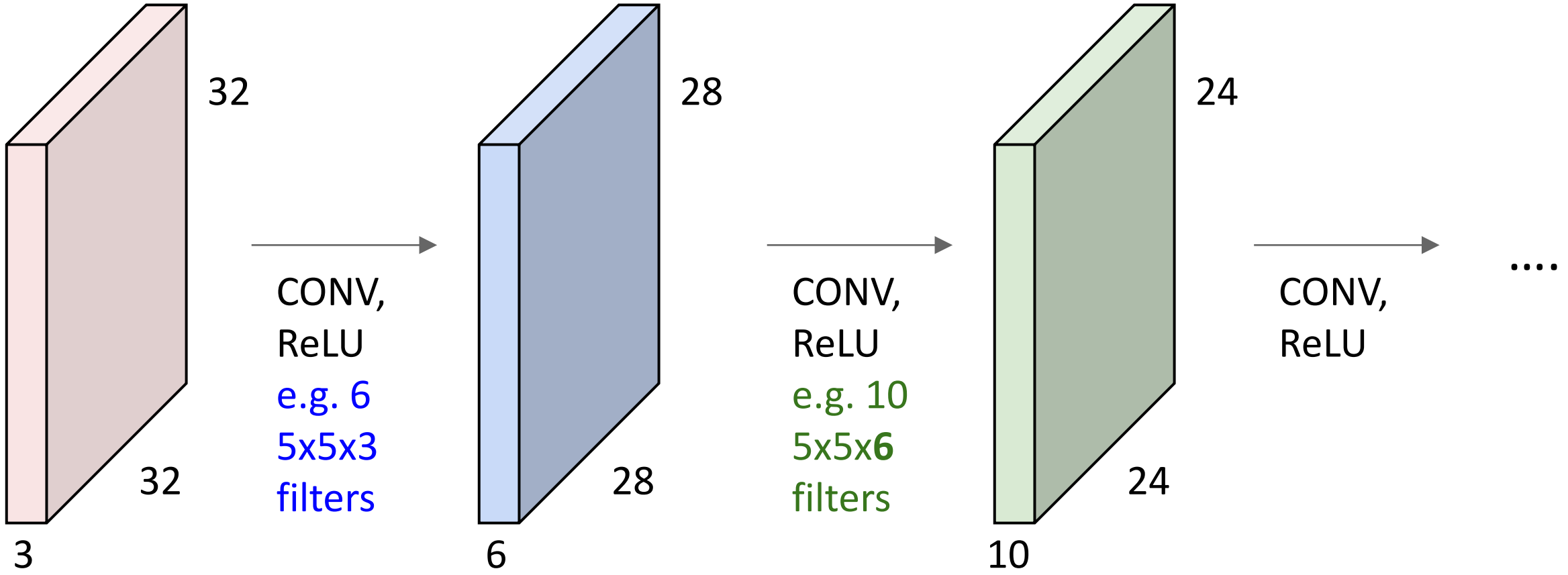
# Convolutional Layers




**1 number:**
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

# Convolutional Architectures



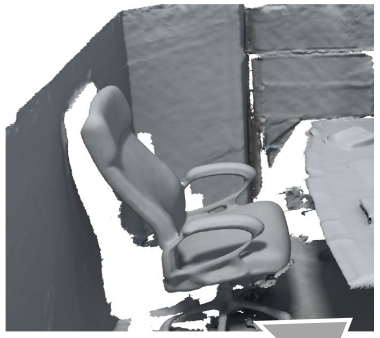

ConvNets are a sequence of convolutional layers, interspersed with activation functions

# Convolutional Neural Networks

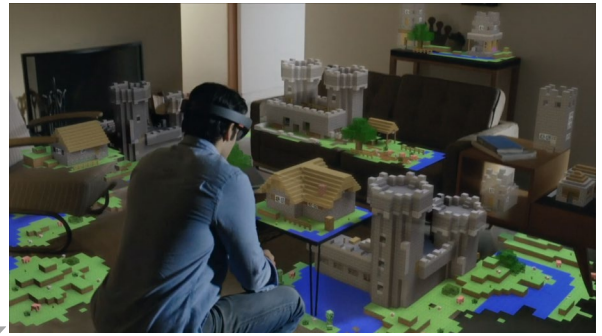
- Filters are doing pattern matching

Image Credits: Yan LeCun

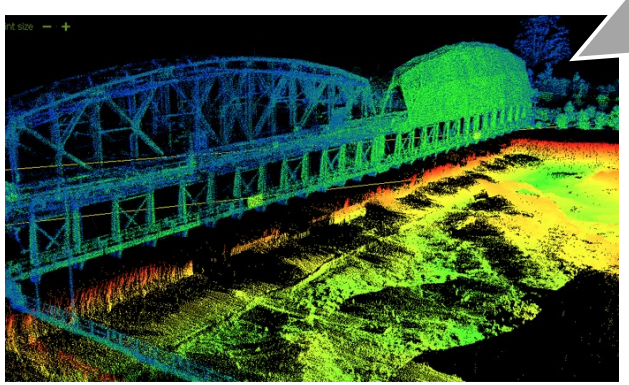
# 3D Applications

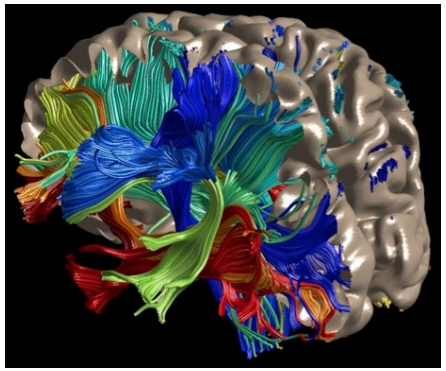
Robotics



Augmented Reality



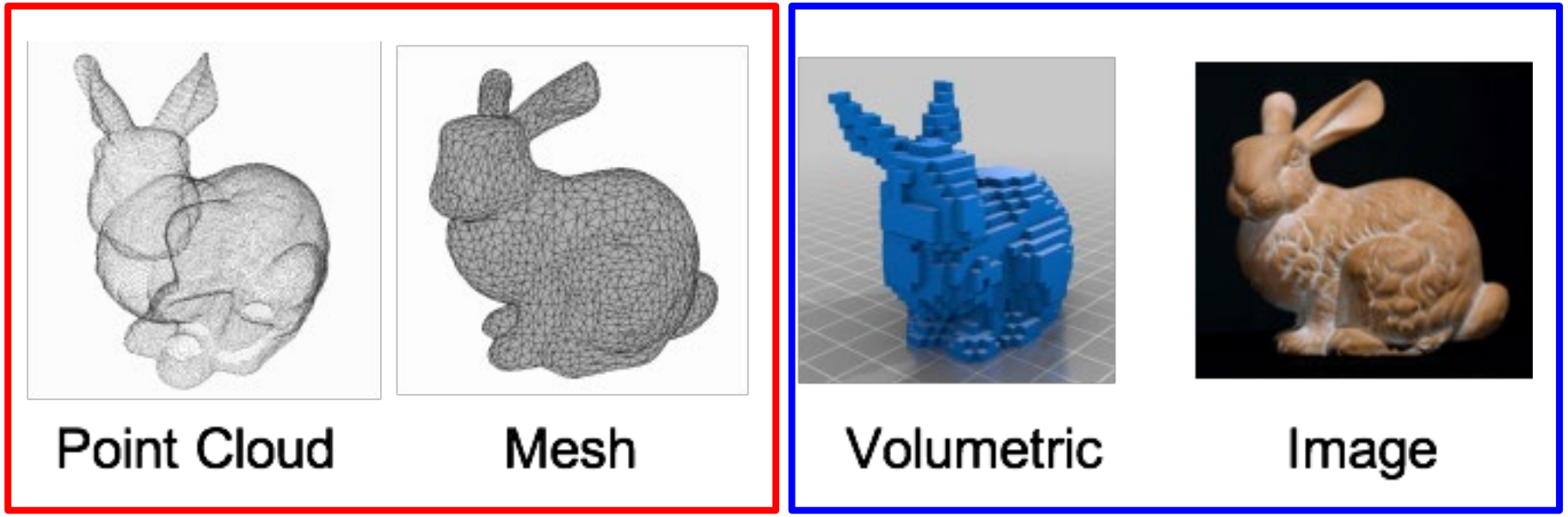
Autonomous driving



Medical Image Processing

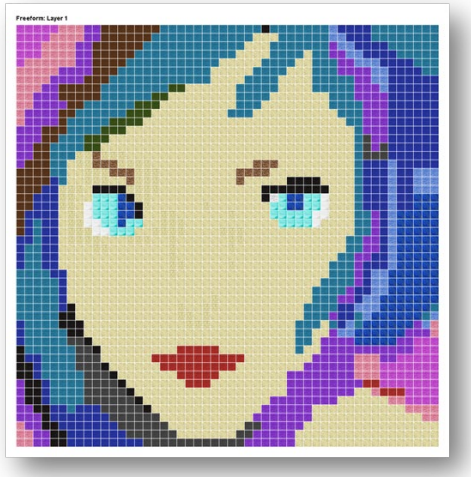
# 3D Representations

- Design good 3D representations for NN to consume




Irregular

Regular



# Multi-view Representation


+ Powerful 2D CNNs

# Multi-view CNNs: Classification

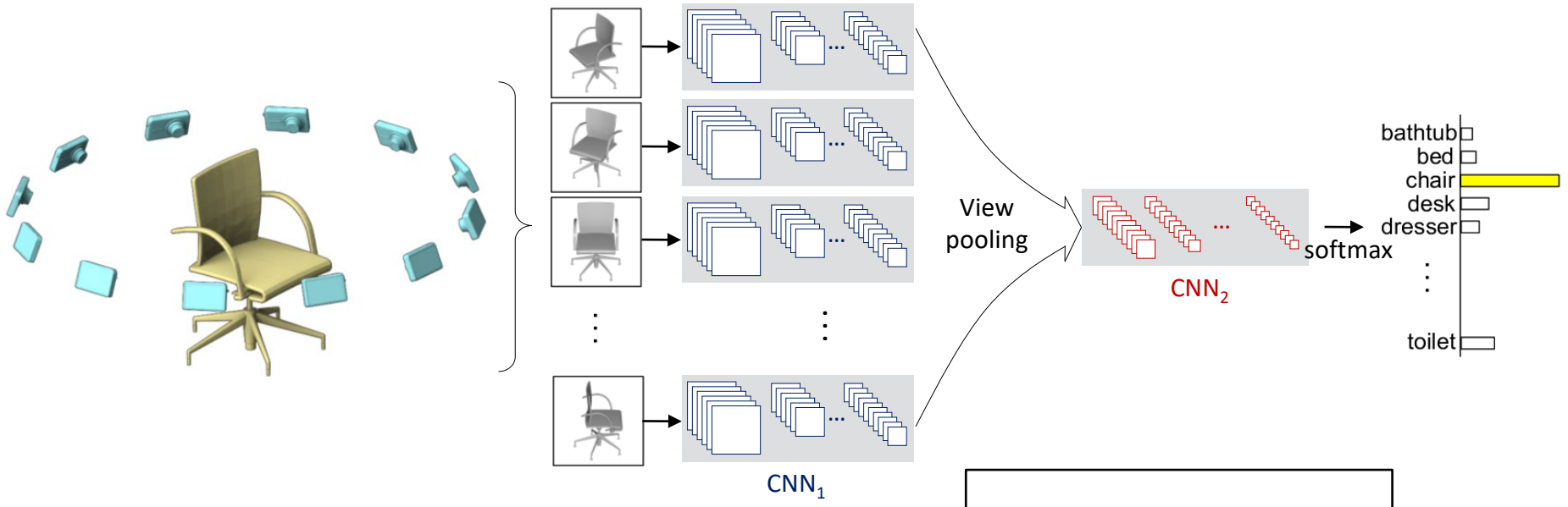
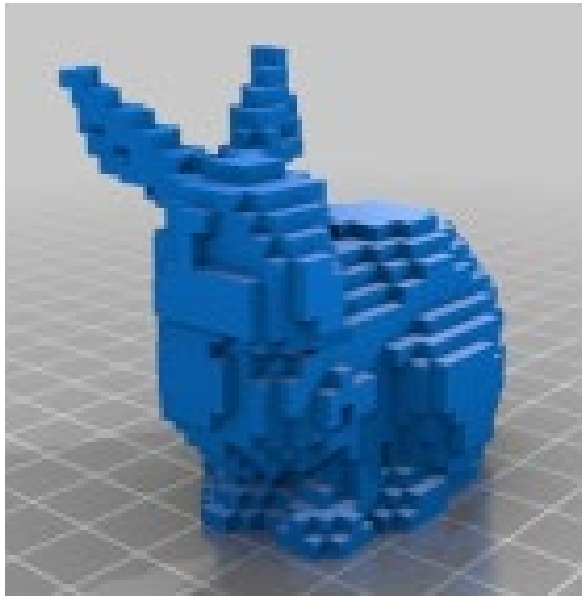



Image Credits: Hang Su

Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-Miller, "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *Proceedings of ICCV 2015*

# Volumetric Representation



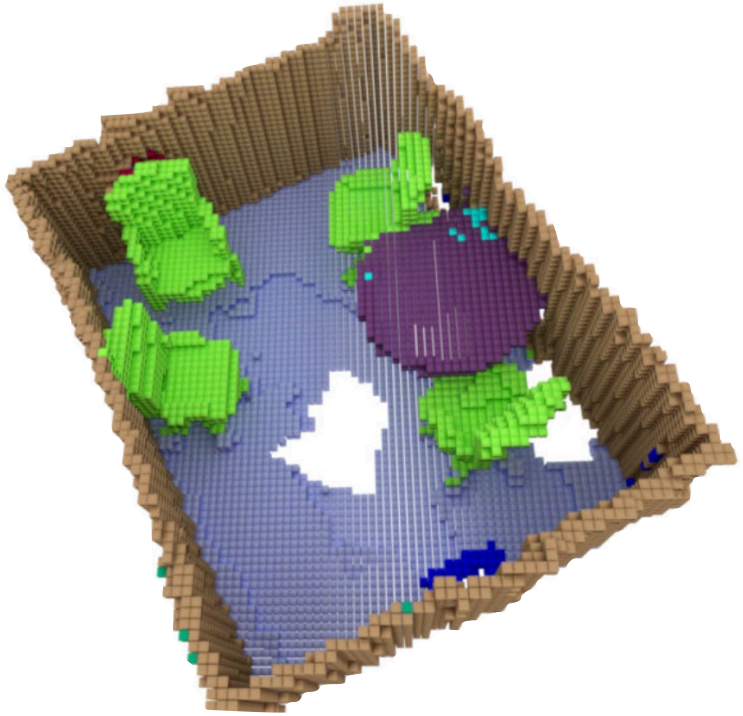
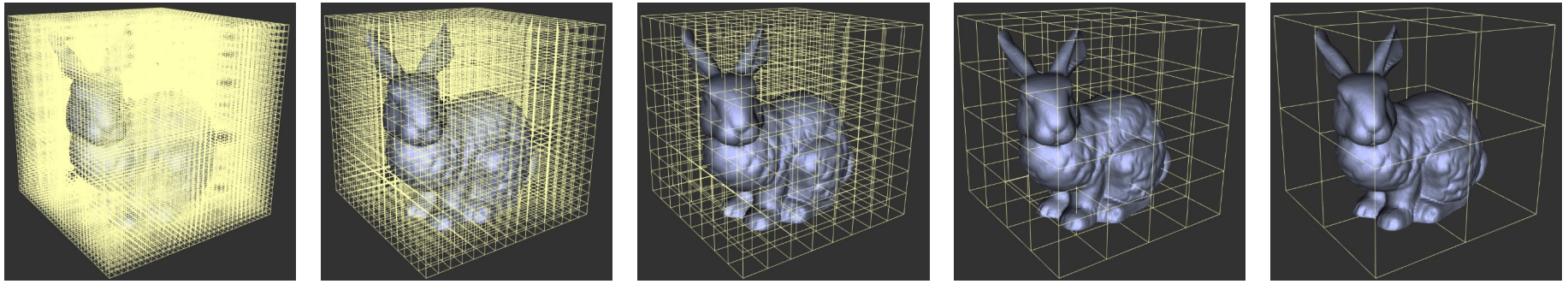
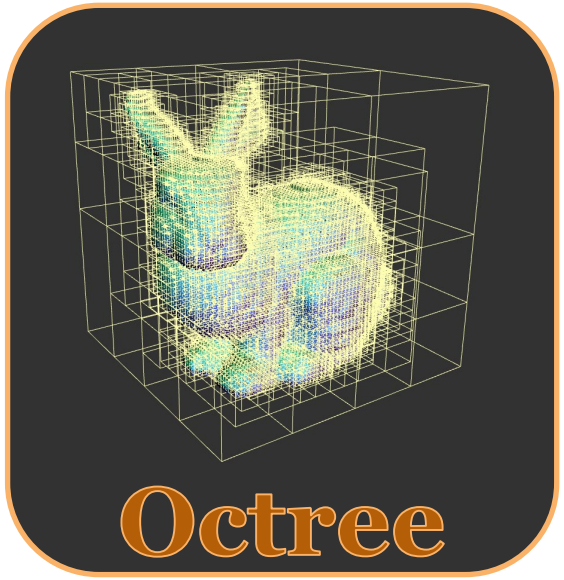


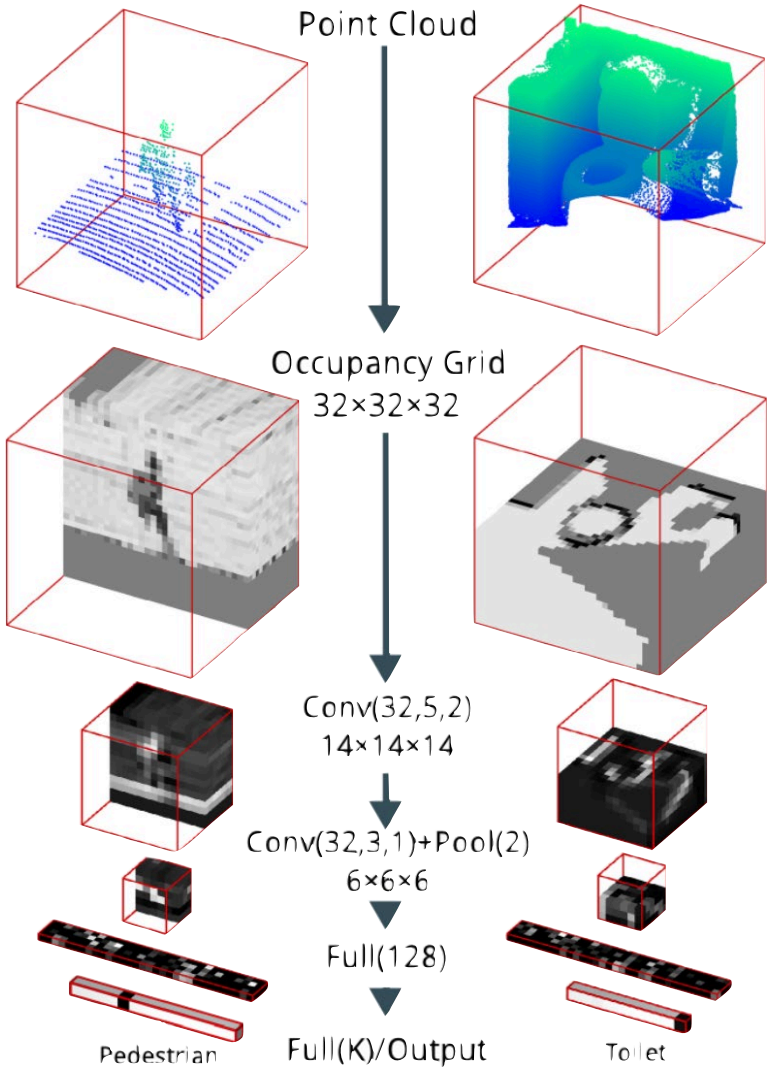
Image Credits: Scannet

# Sparse Volumetric Representation





# Volumetric Representation: Classification



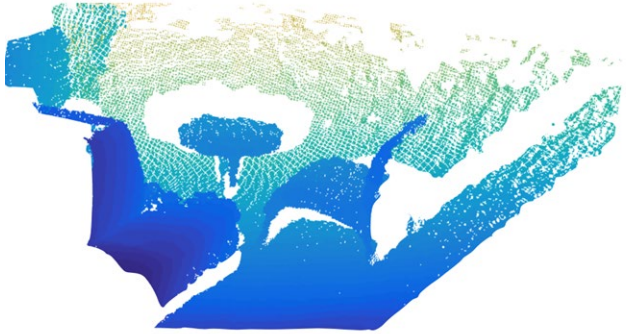

Daniel Maturana and Sebastian Scherer, **"VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition",** *IROS2015*

# Conclusion

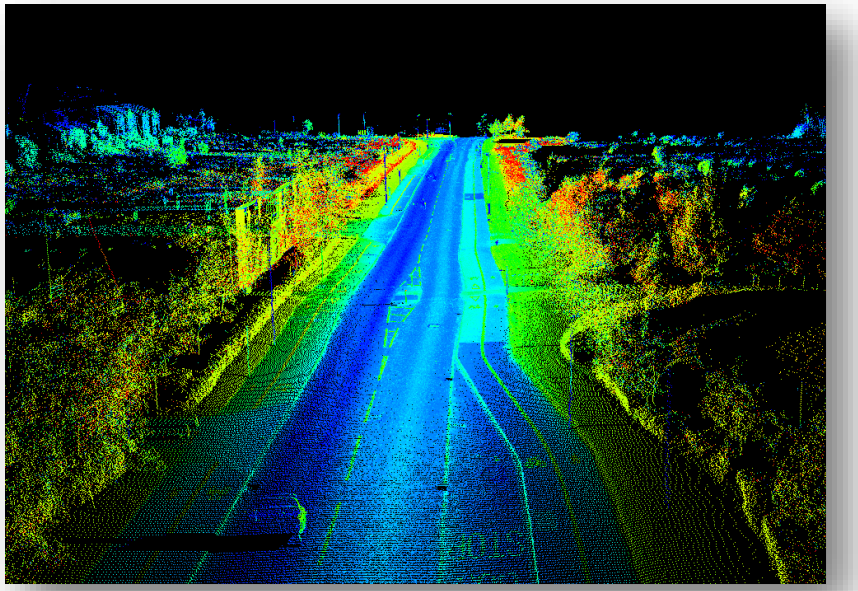
- 3D deep learning is a new and active research direction
- 3D data have different representation
- Multi-view CNNs take advantage of the state-of-the-art 2D CNNs
- Volumetric 3D CNNs suffer from the “curse of dimensionality”

# Today: Deep Learning on Point Cloud Data

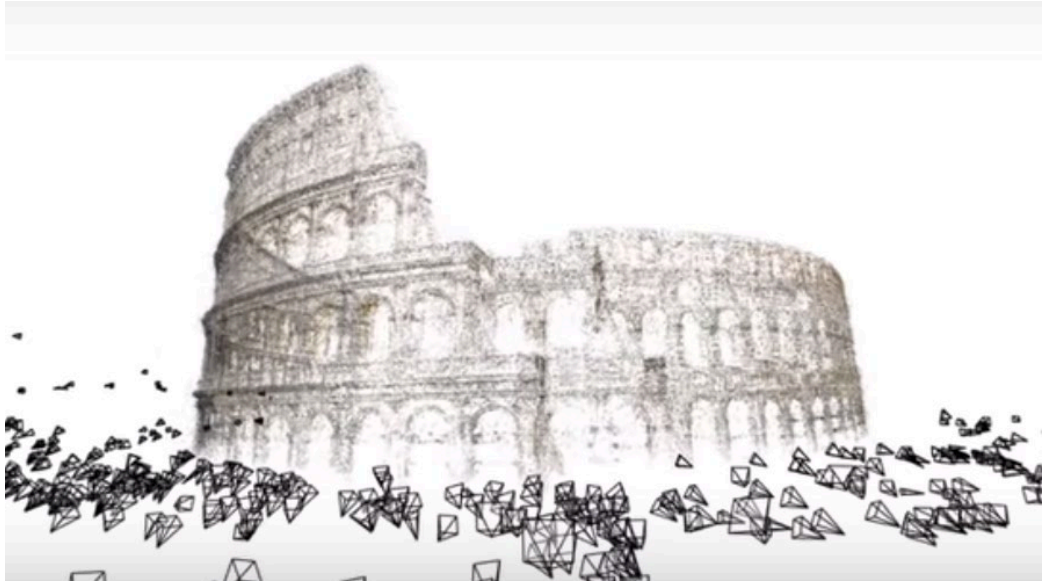
Non-regular 3D data



# Point Clouds are Commonplace



Lidar point clouds (LizardTech)

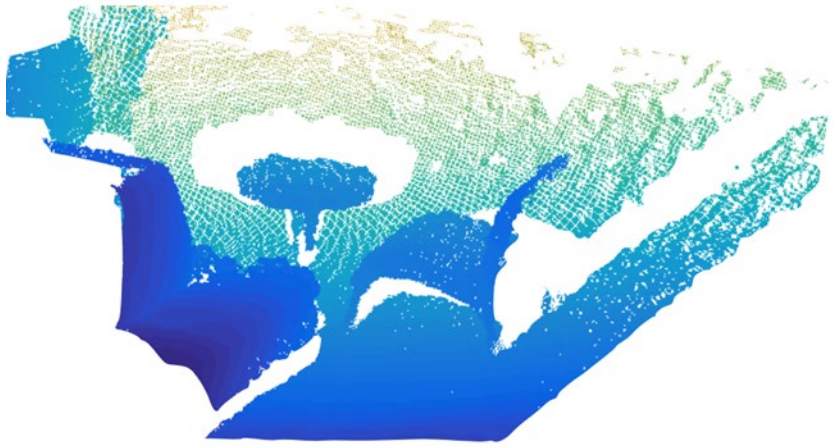


Structure from motion (Microsoft)

Depth camera (Intel)

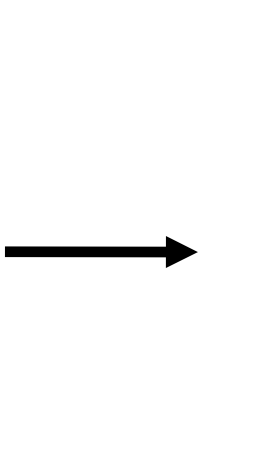
# Deep Nets on 3D Point Cloud Data

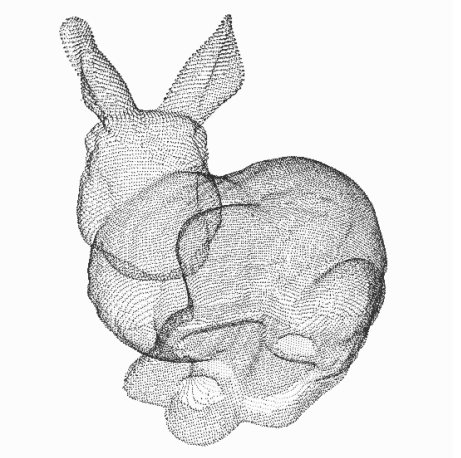
- Close to raw sensor data
- Representationally simple
- Irregular




LiDAR

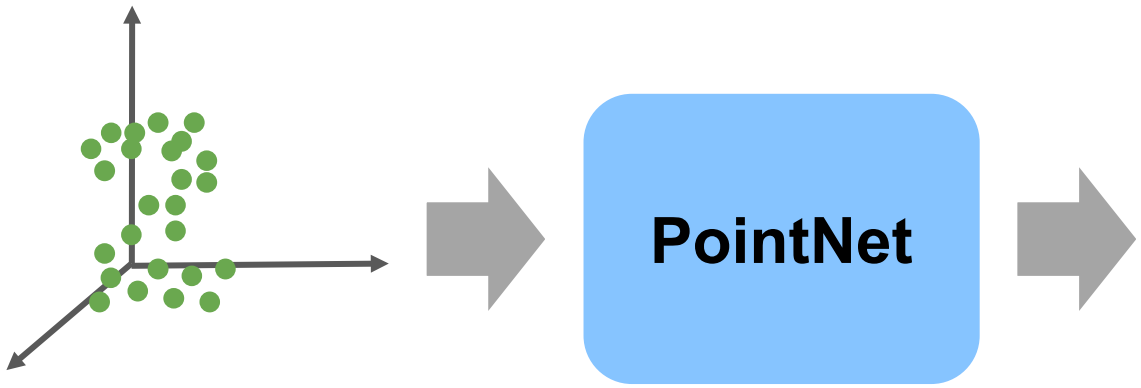
Depth Sensor





**Point Cloud**

# Deep Nets for PCs: PointNet and PointNet++




*Object Classification*

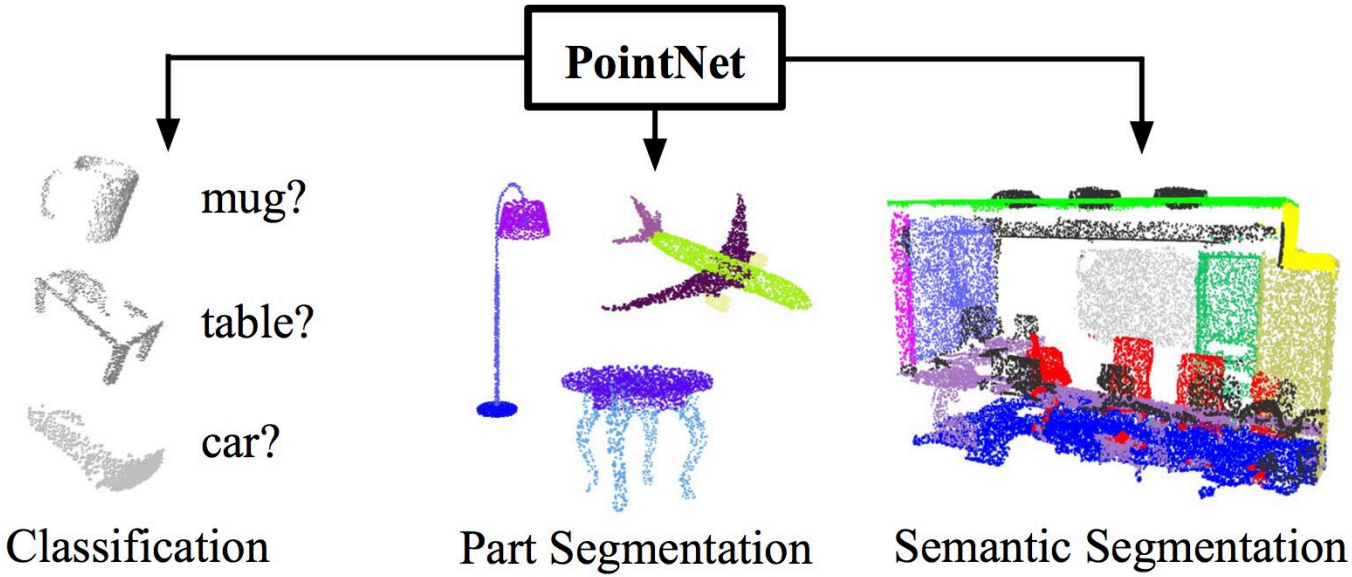
*Object Part Segmentation*

*Semantic Scene Parsing*

*...*

**End-to-end learning** for irregular point data

**Unified** framework for various tasks




Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. (CVPR'17)

# Invariances

*The model has to respect key desiderata for point clouds:*

**Point Permutation Invariance**

Point cloud is a set of unordered points

**Spatial Transformation Invariance**

Point cloud rigid motions should not alter classification results

**Sampling Invariance**

Output a function of the underlying geometry and not the sampling

# Permutation Invariance: Symmetric Functions

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

**Examples:**

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

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

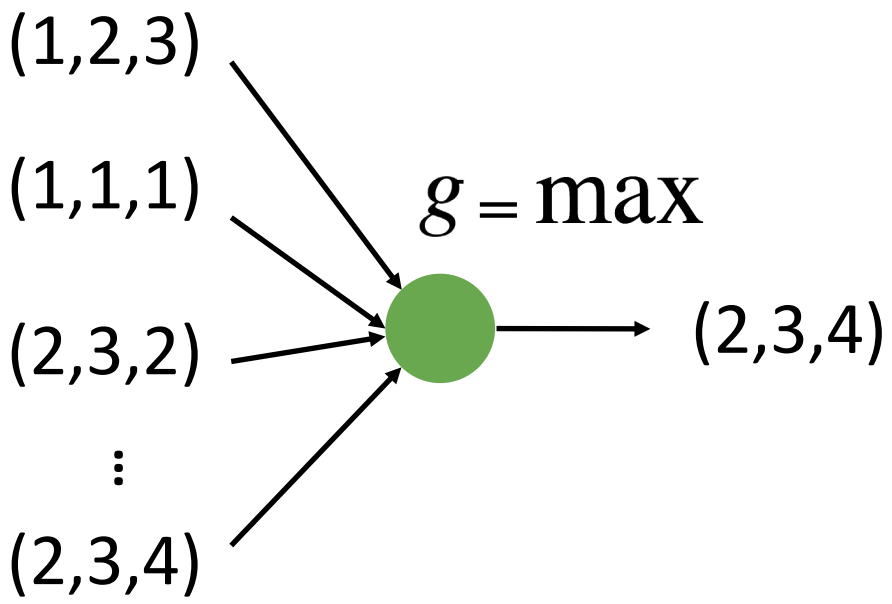
…

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

# Construct Symmetric Functions by Neural Networks

Simplest form: directly aggregate all points with a symmetric operator $g$

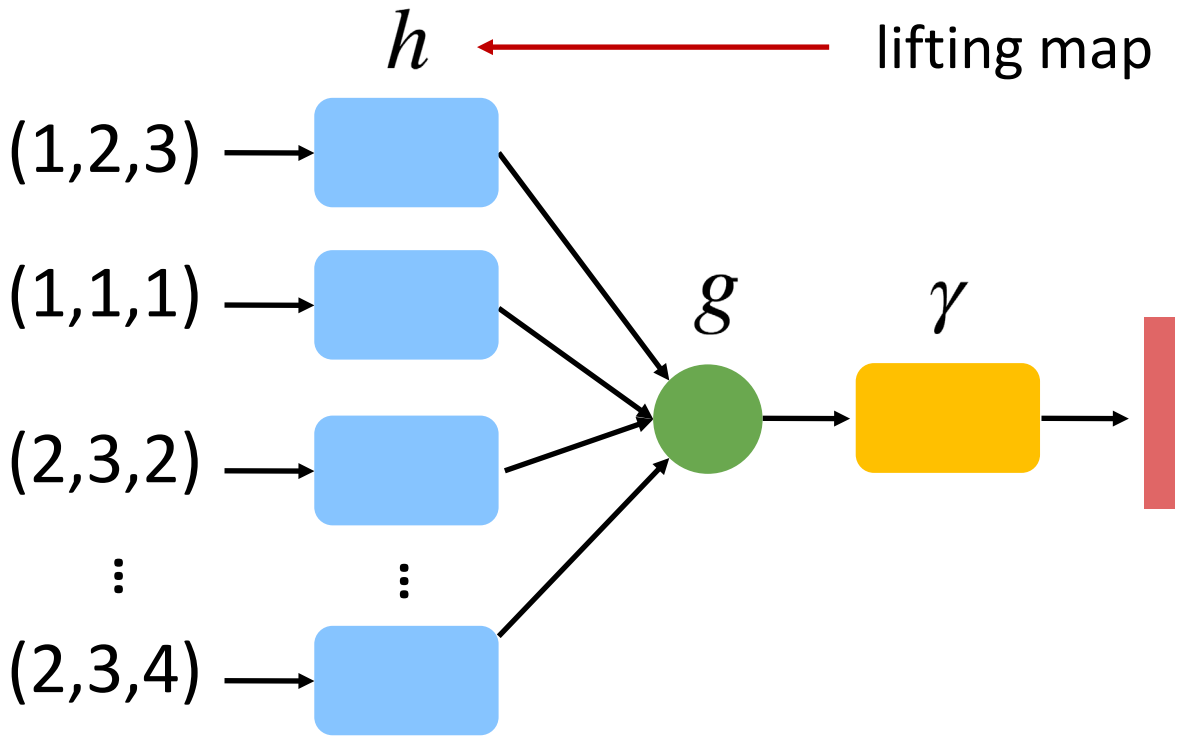
**Just discovers simple extreme/aggregate properties of the geometry.**



# Construct Symmetric Functions by Neural Networks

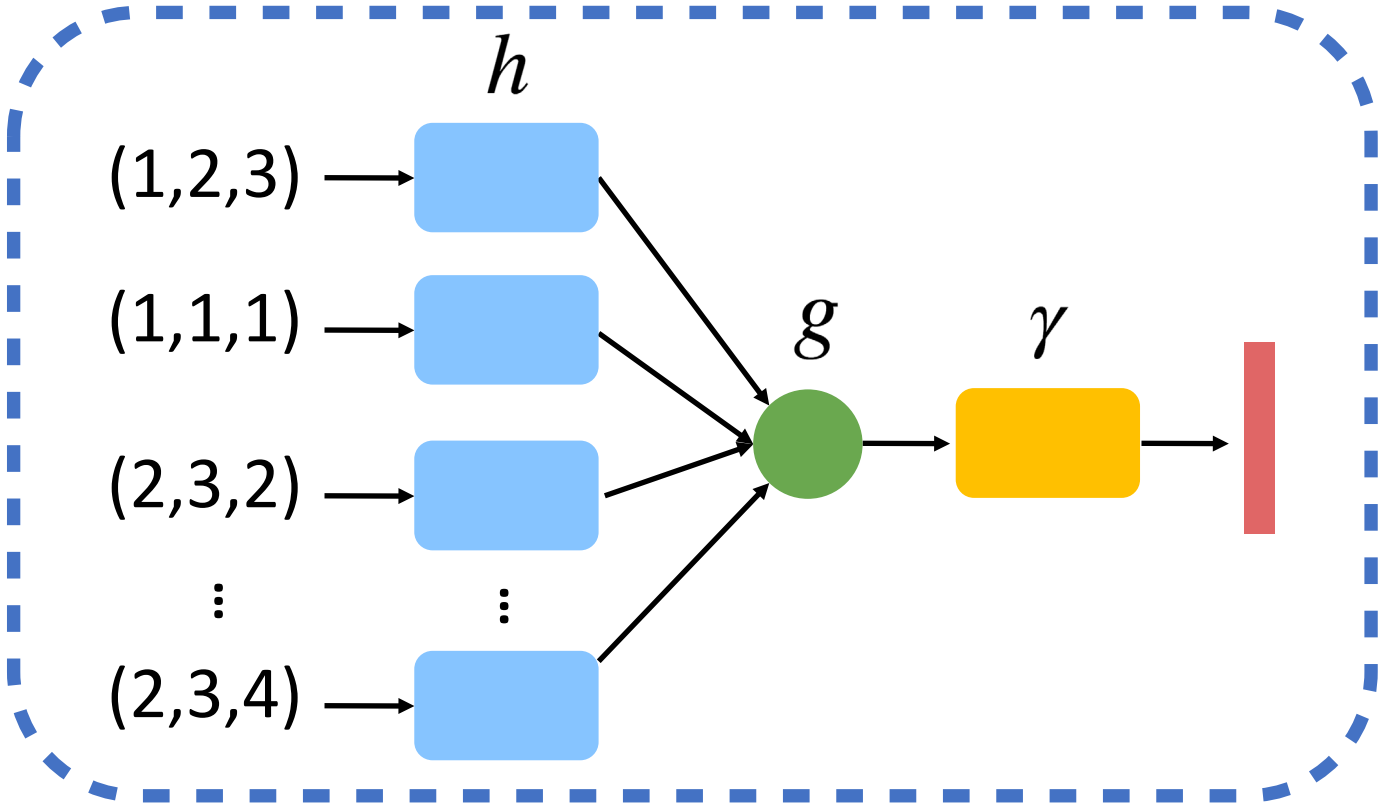
Embed points in a high-dim space before aggregation.

**Aggregation in the (redundant) high-dim space encodes more interesting properties of the geometry.**



# Construct Symmetric Functions by Neural Networks

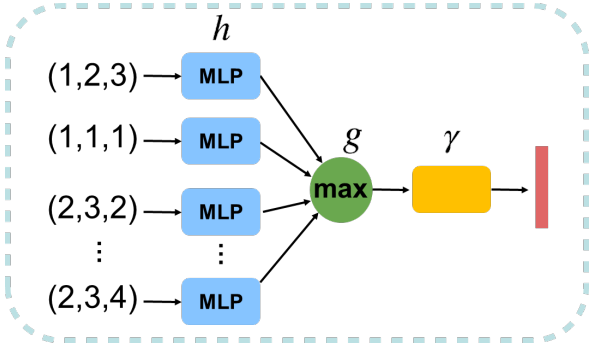
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if $g$ is symmetric




$h$ lifts points to a high-dimensional space

**PointNet** (vanilla)

# Symmetric Functions: Polynomials




$$2\sum_{i\neq j} x_i x_j = (\sum_i x_i)^2 - \sum_i x_i^2 \qquad \sum_{i\neq j}(x_i - x_j)^2 = 3\sum_i x_i^2 - (\sum_i x_i)^2$$

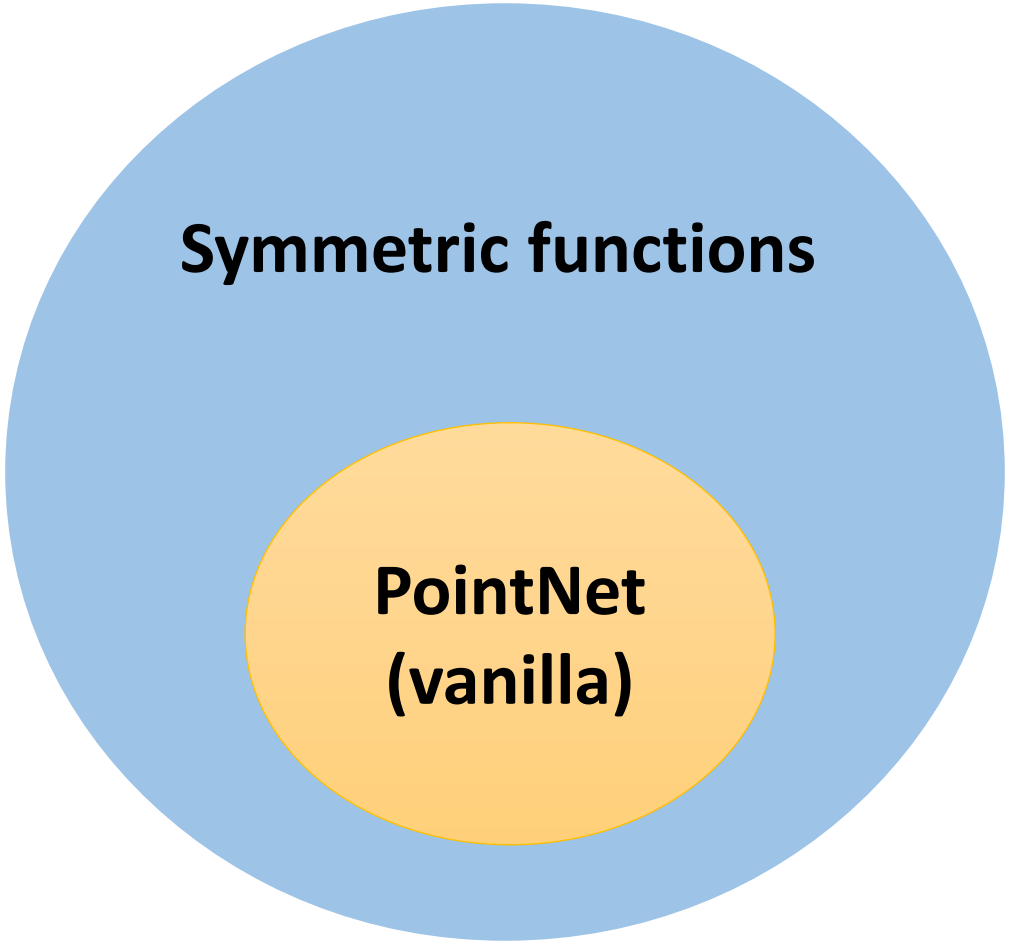
- In fact, any symmetric polynomial in the $x_i$ can be expressed as a polynomial in sums of the form

$$\sum_i x_i^k$$

and can be computed by

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

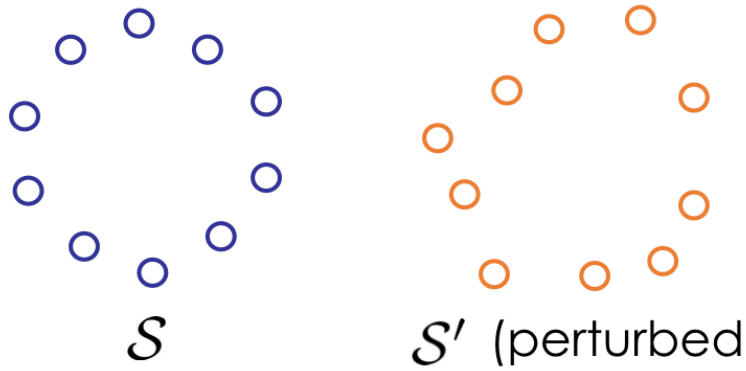
# What Symmetric Functions Can Be Constructed By PointNet?



# PointNet as a Universal Approximation to Set Functions

**Hausdorff continuous:**

$f: 2^x \to \mathbb{R}$ is a continuous set function w.r.t Hausdorff distance




if $d_{Hausdorff}(S, S') \approx 0$, then $f(S) \approx f(S')$

**Theorem**

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

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

$S \subseteq R^d$

**PointNet (vanilla)**

Voxel occupancy maps

# Invariances

*The model has to respect key desiderata for point clouds:*

**Point Permutation Invariance**

Point cloud is a set of unordered points

**Spatial Transformation Invariance**

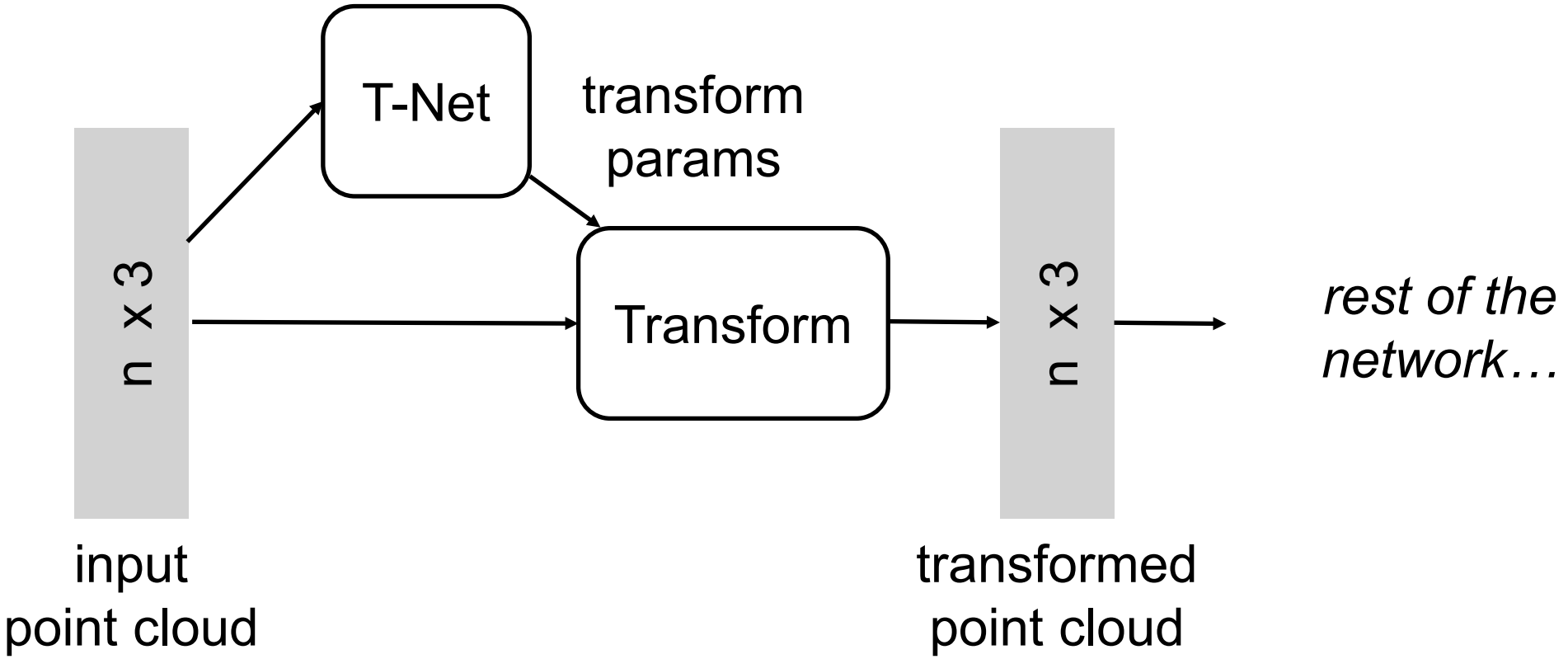
Point cloud rigid motions should not alter classification results

**Sampling Invariance**

Output a function of the underlying geometry and not the sampling

# Input Alignment by Transformer Network

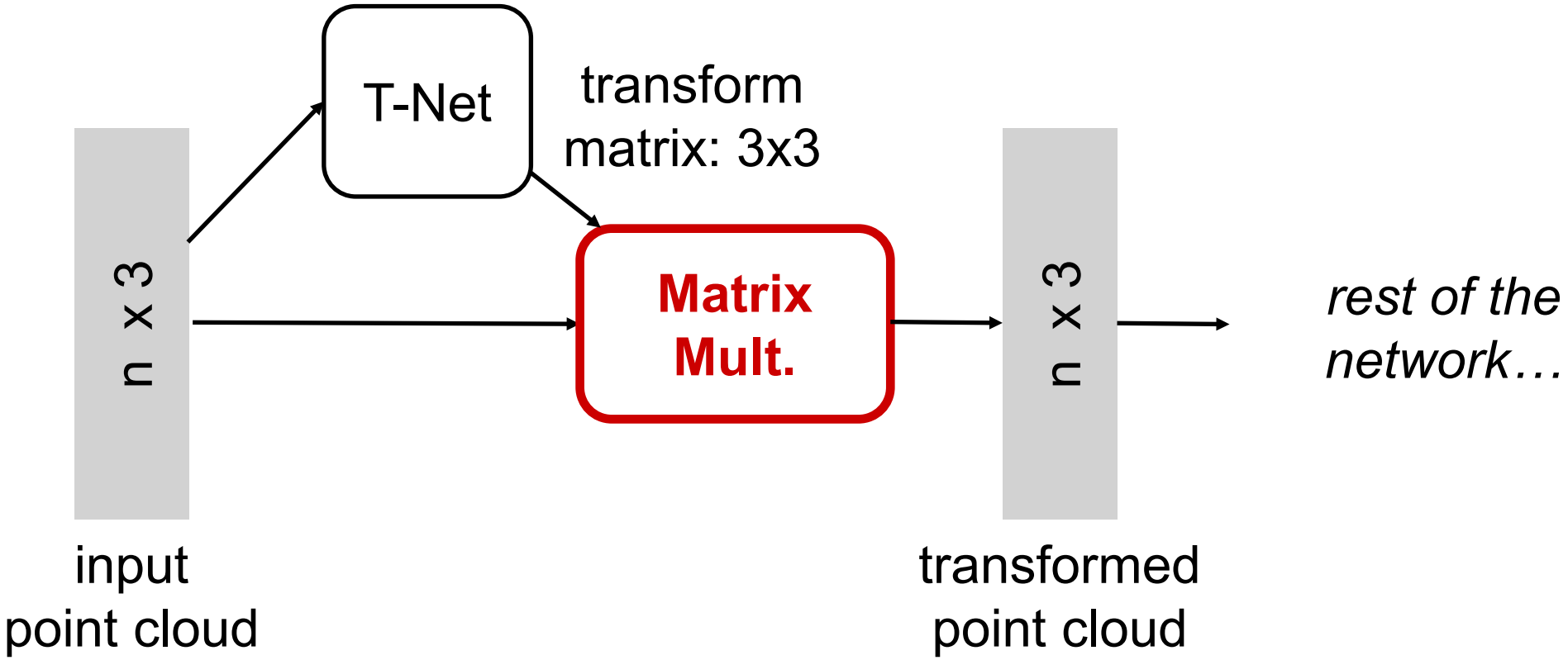
Idea: Data dependent transformation for automatic alignment



# Input Alignment by Transformer Network

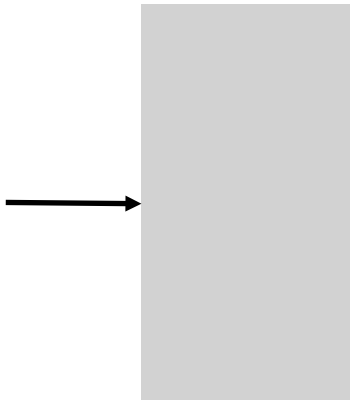
Idea: Data dependent transformation for automatic alignment

The transformation is just matrix multiplication!



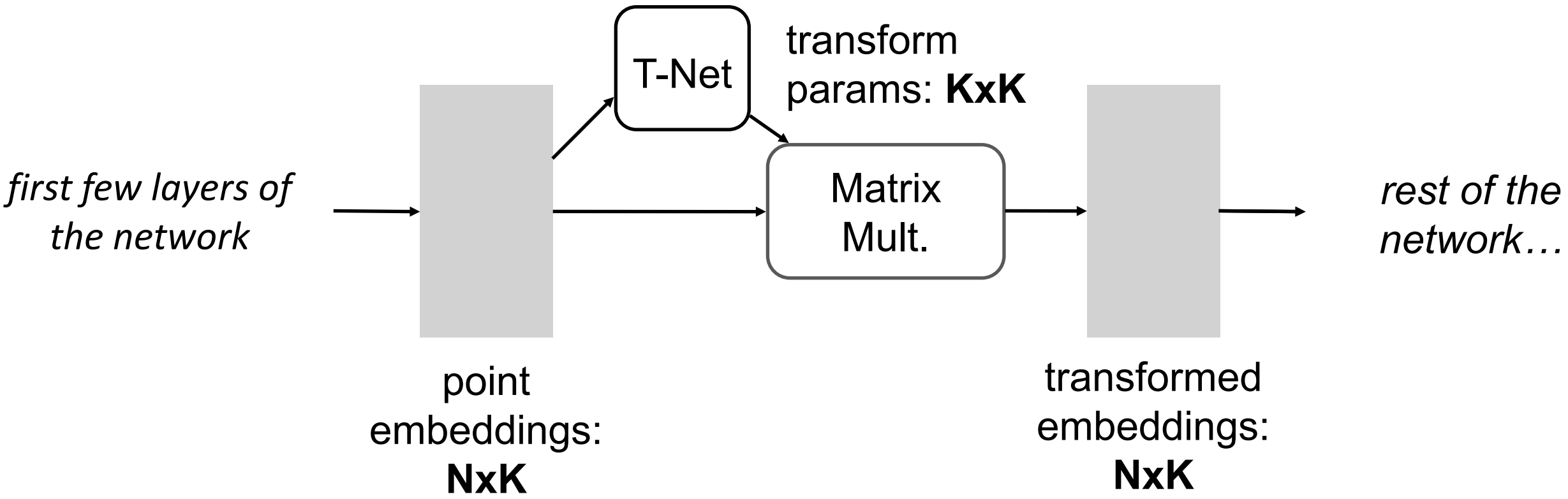
# Embedding Space Alignment

*first few layers of the network*

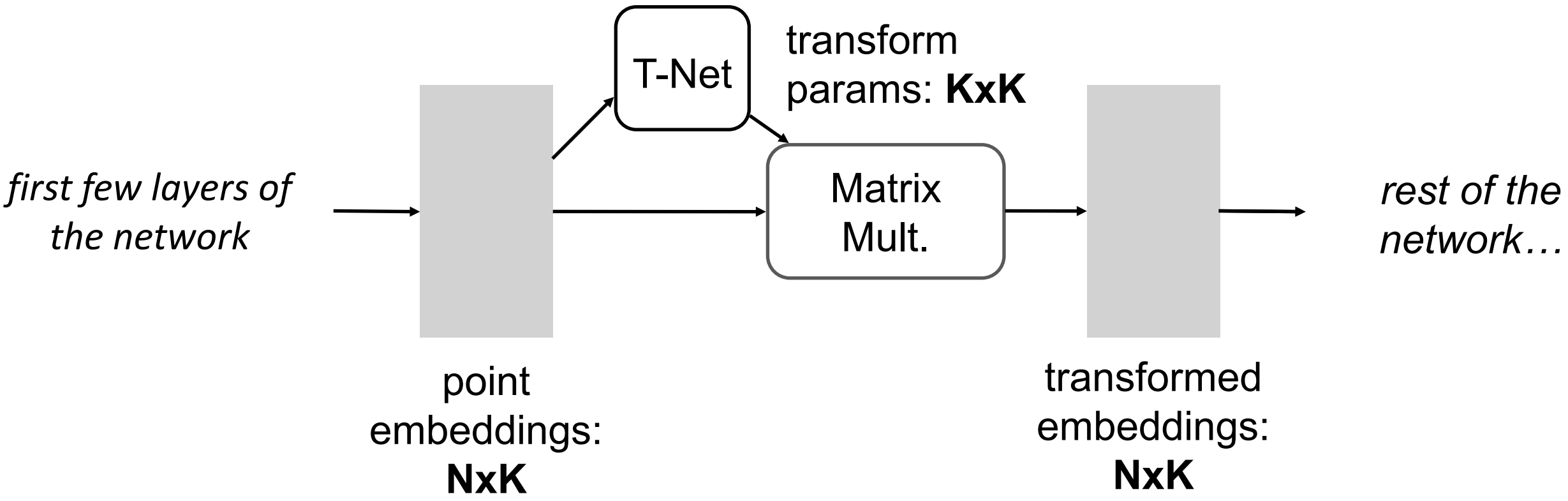


point embeddings:
**NxK**

# Embedding Space Alignment



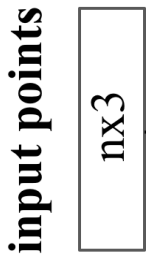
# Embedding Space Alignment



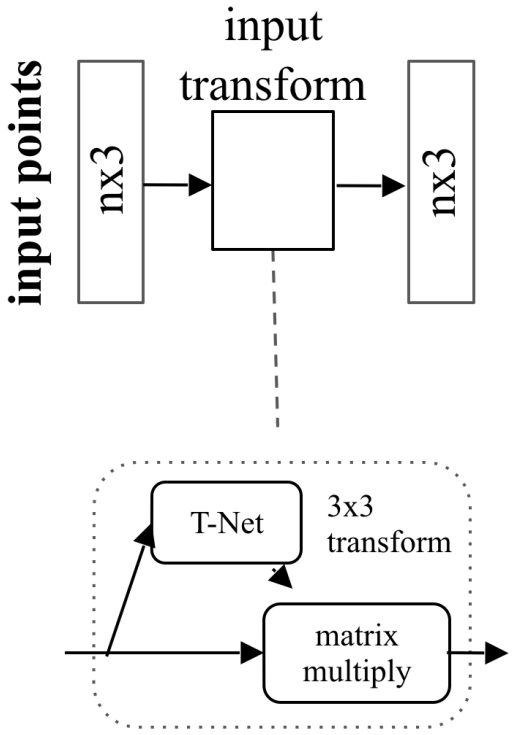

**Regularization loss:**

Transform matrix close to orthogonal: $L_{reg} = \|I - AA^T\|_F^2$

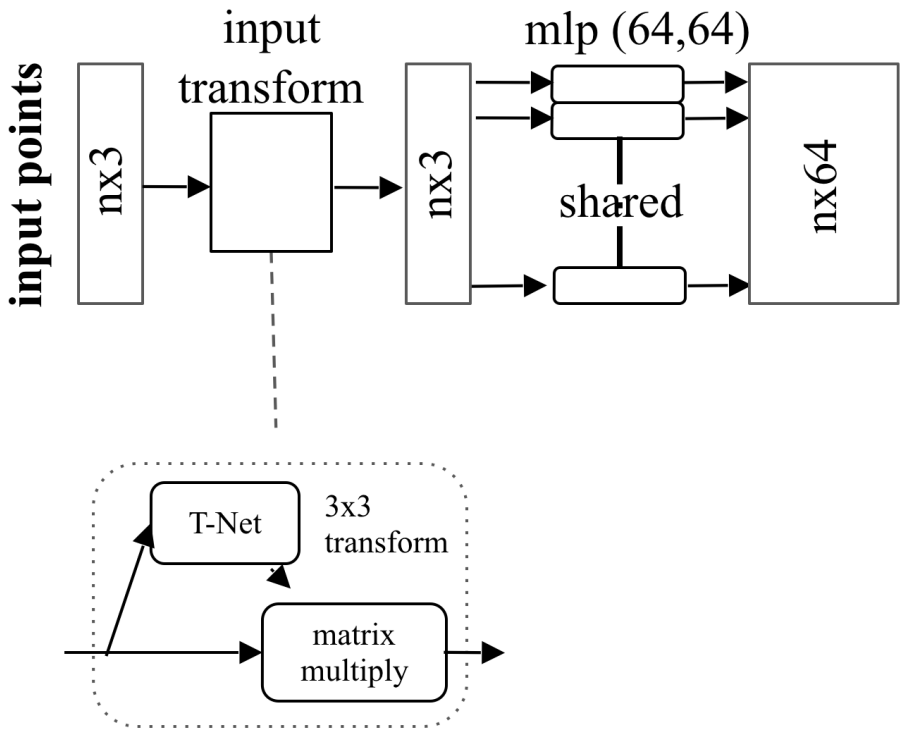
# PointNet Classification Network



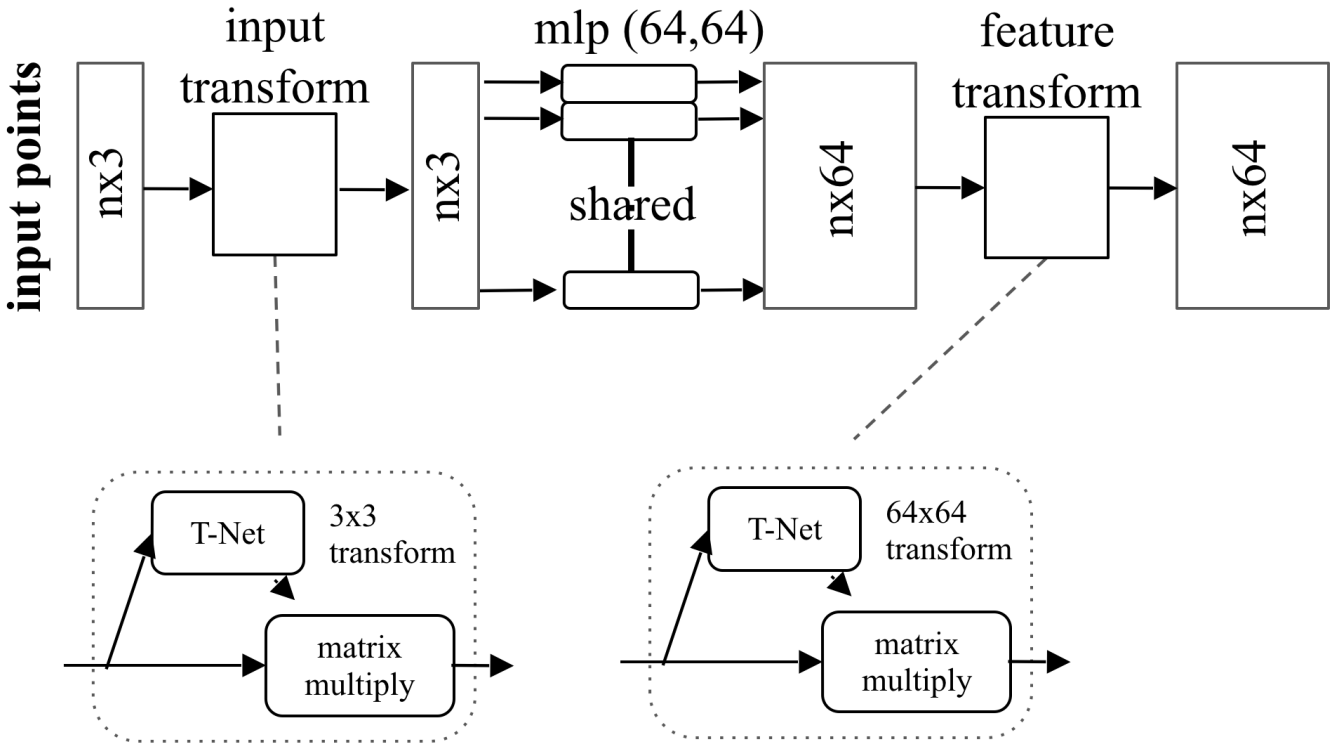
# PointNet Classification Network



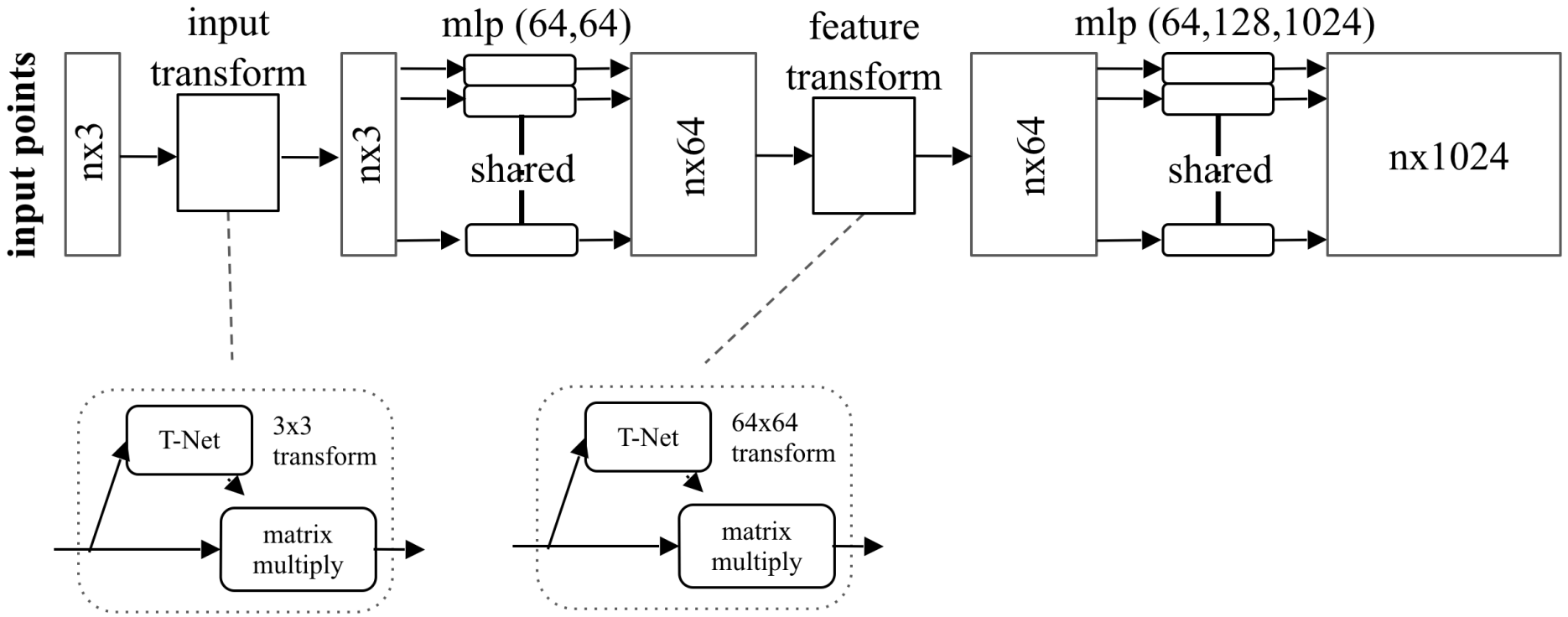
# PointNet Classification Network



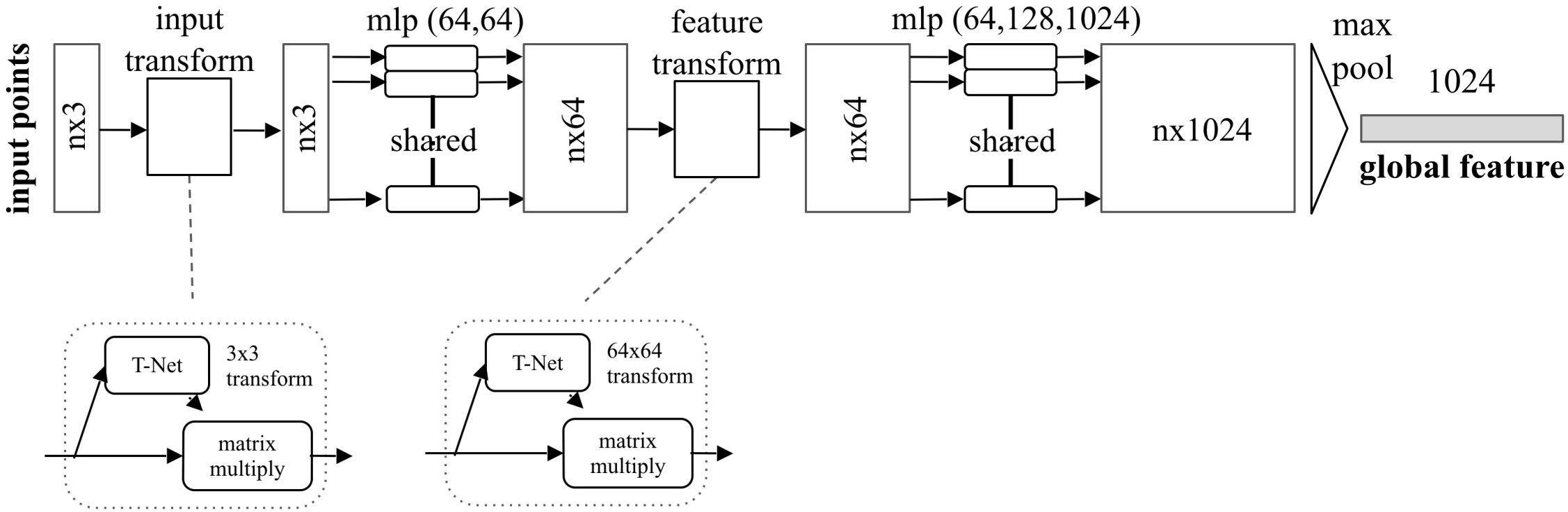
# PointNet Classification Network



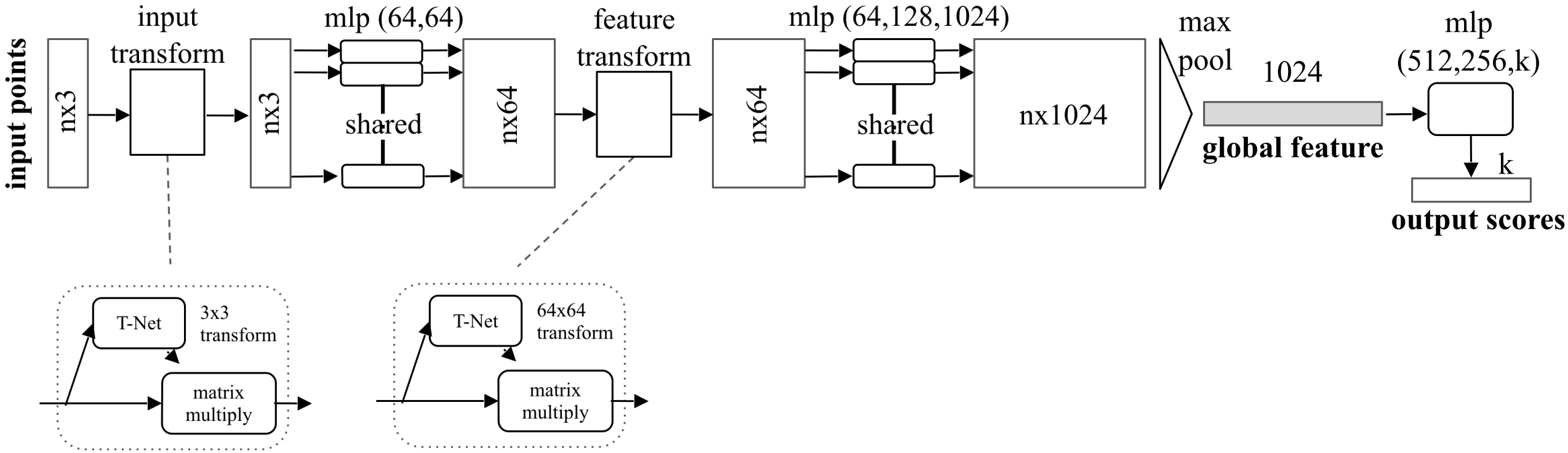
# PointNet Classification Network



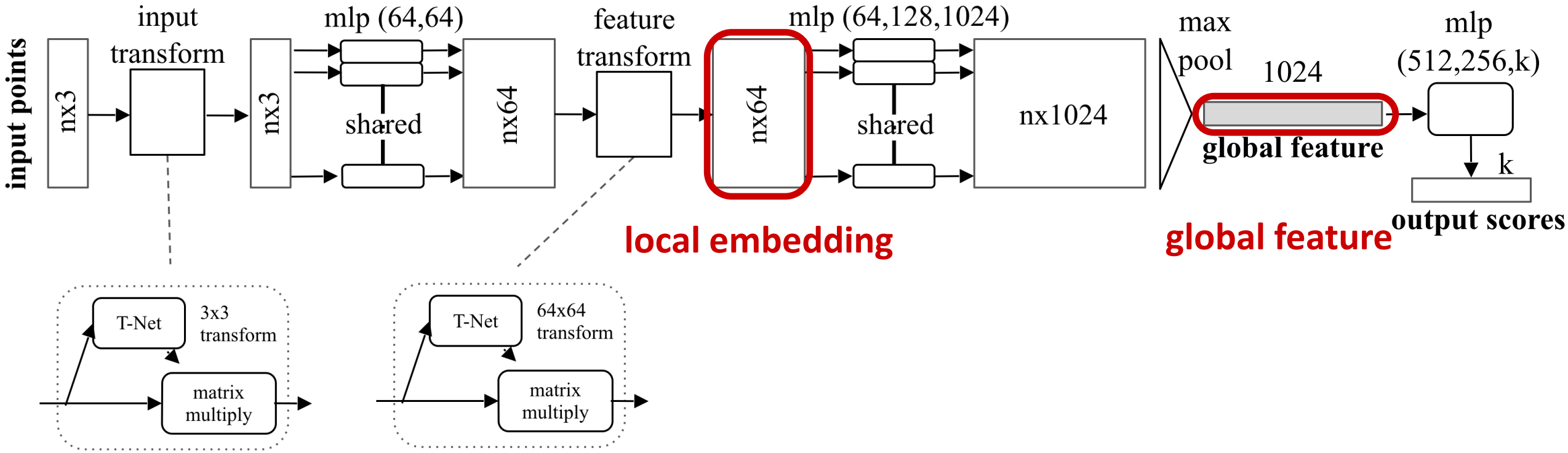
# PointNet Classification Network



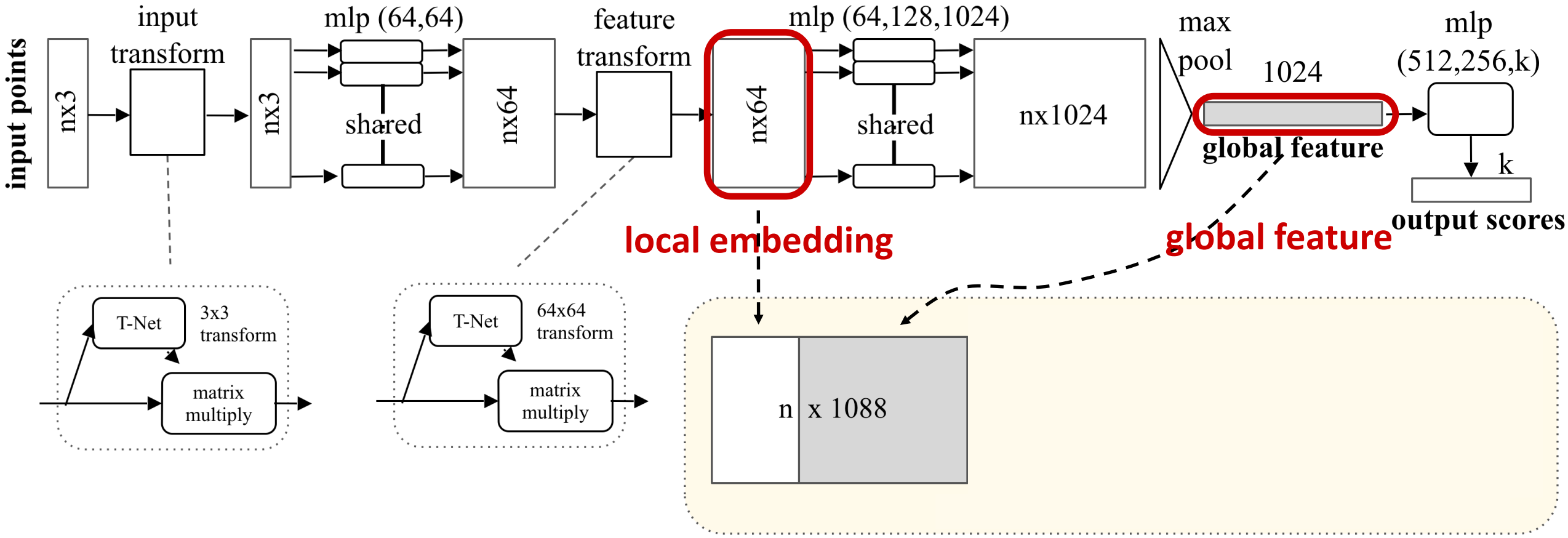
# PointNet Classification Network



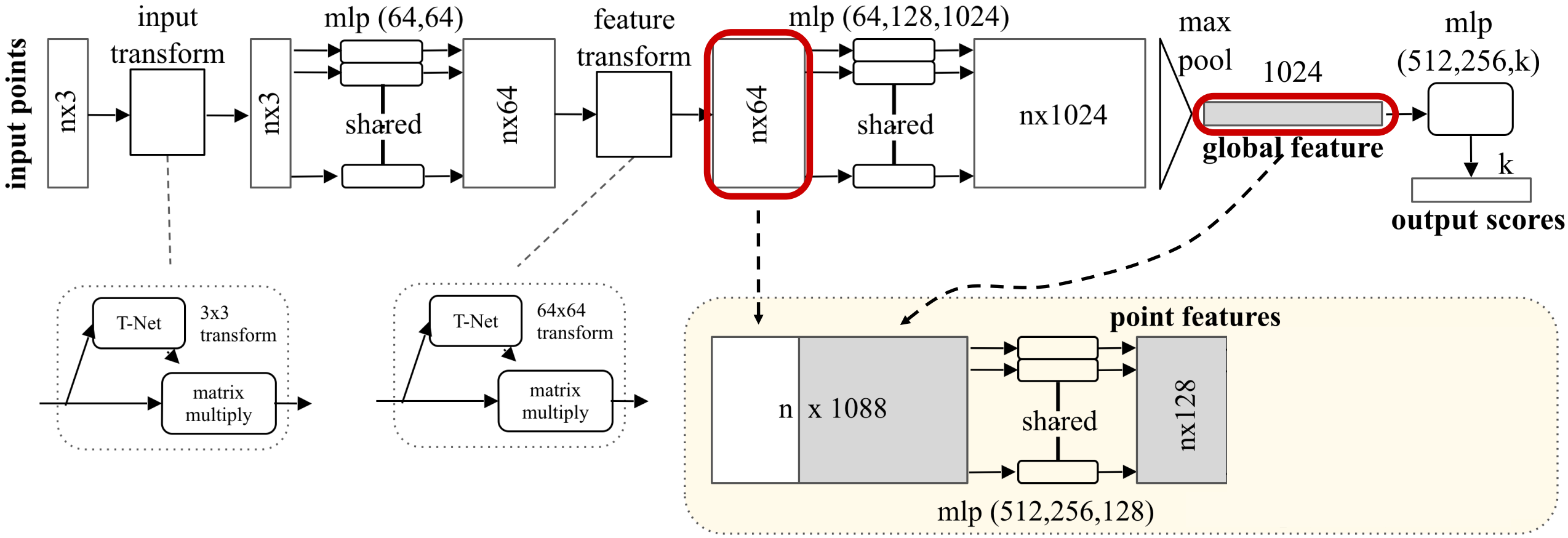
# Extension to PointNet Segmentation Network



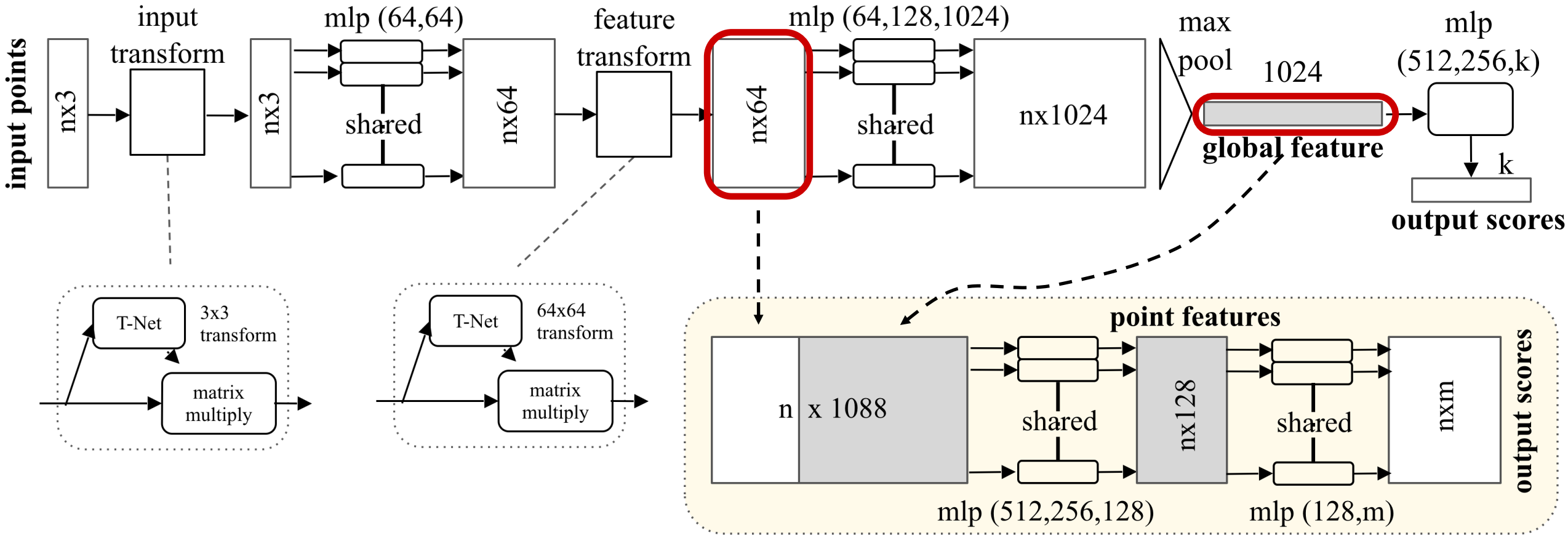
# Extension to PointNet Segmentation Network



# Extension to PointNet Segmentation Network



# Extension to PointNet Segmentation Network



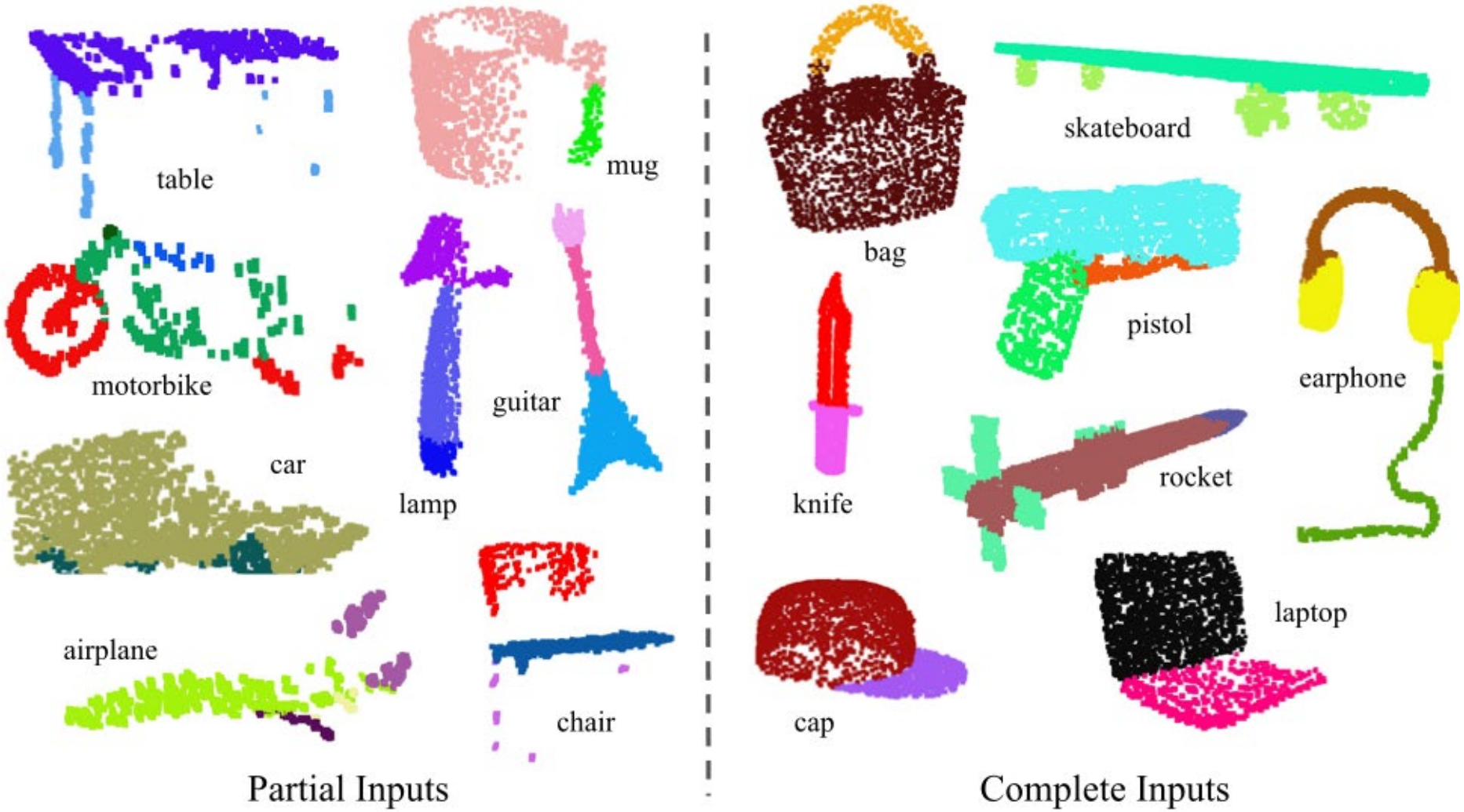
# Results on Object Classification

3D CNNs

| | 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** |

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

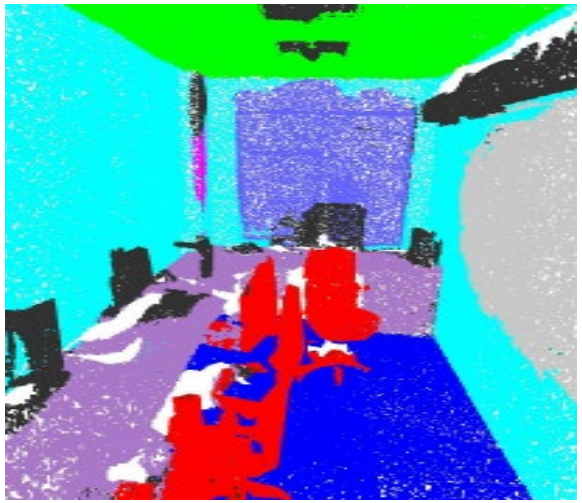
# Results on Object Part Segmentation

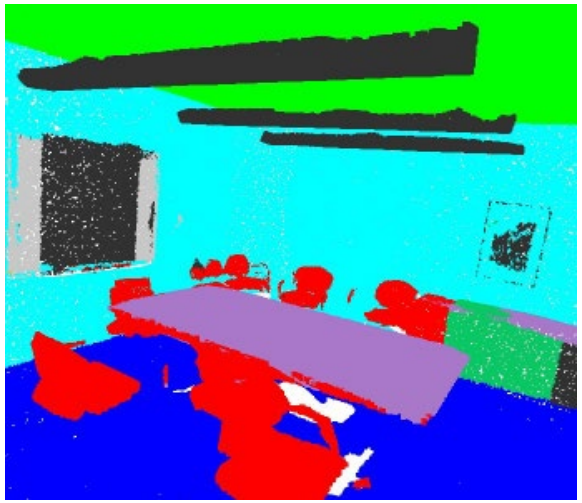


# Results on Semantic Scene Parsing

Input

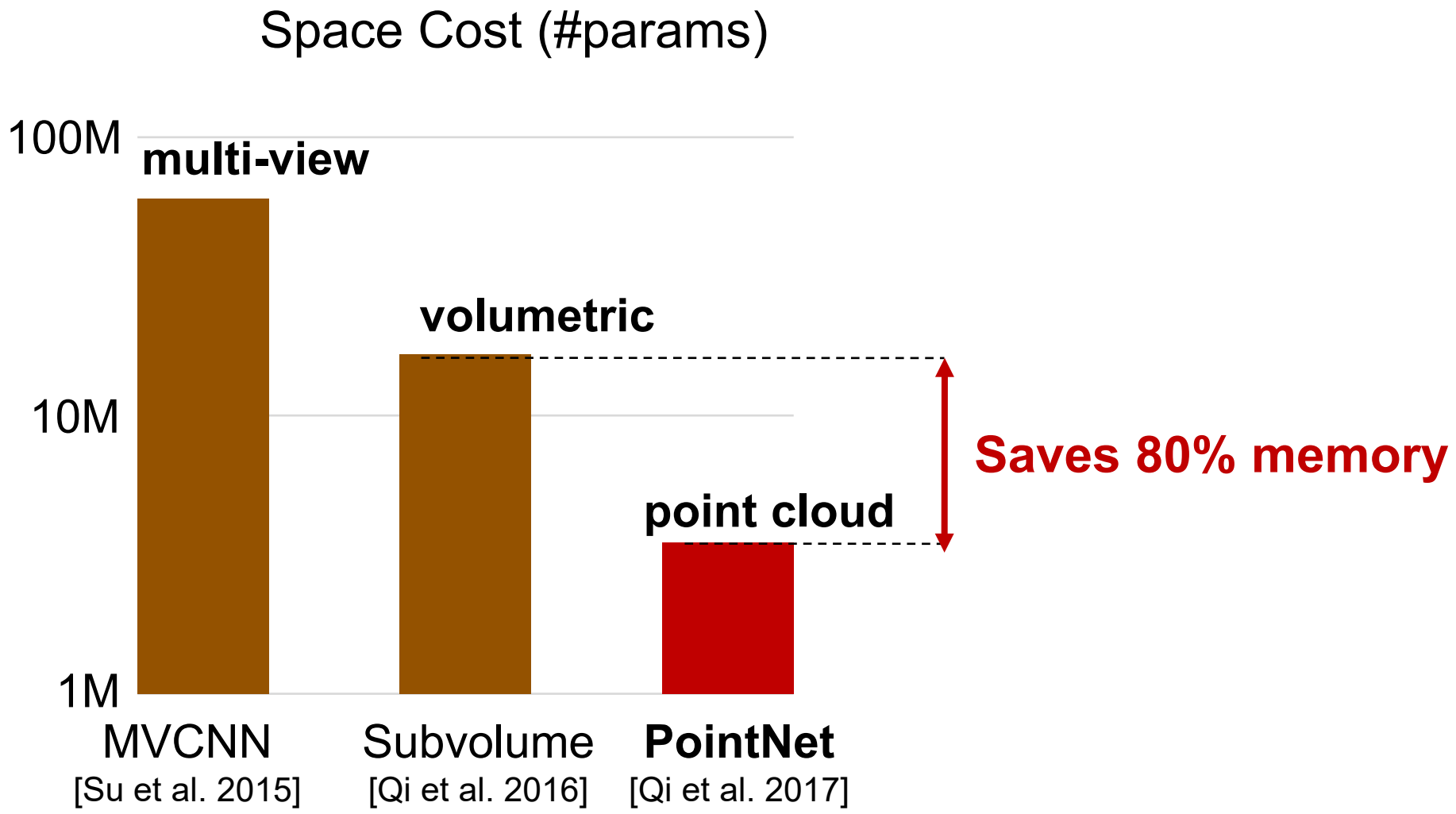
Output



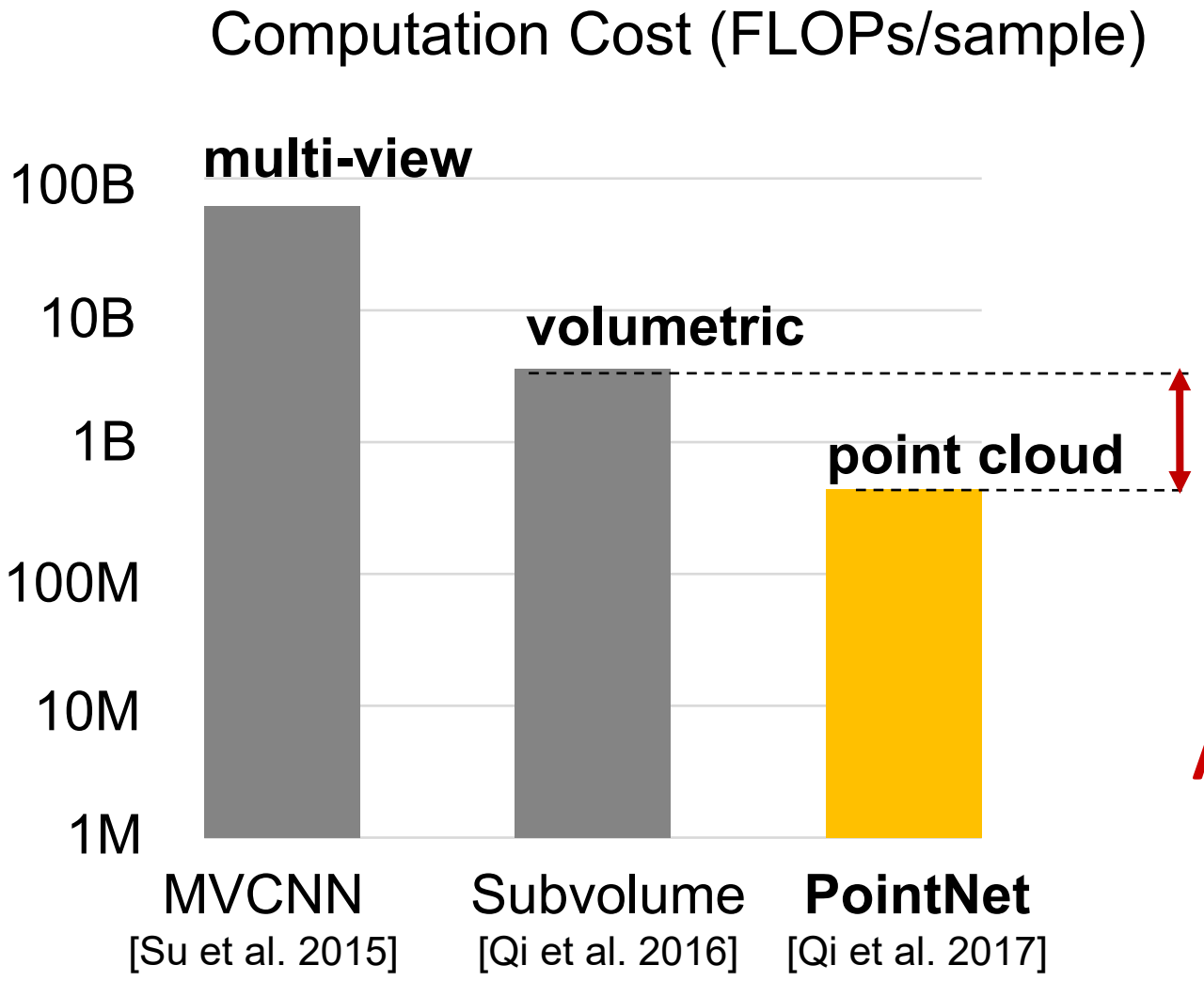


*dataset: Stanford 2D-3D-S (Matterport scans)*

# PointNet is Light-Weight and Fast



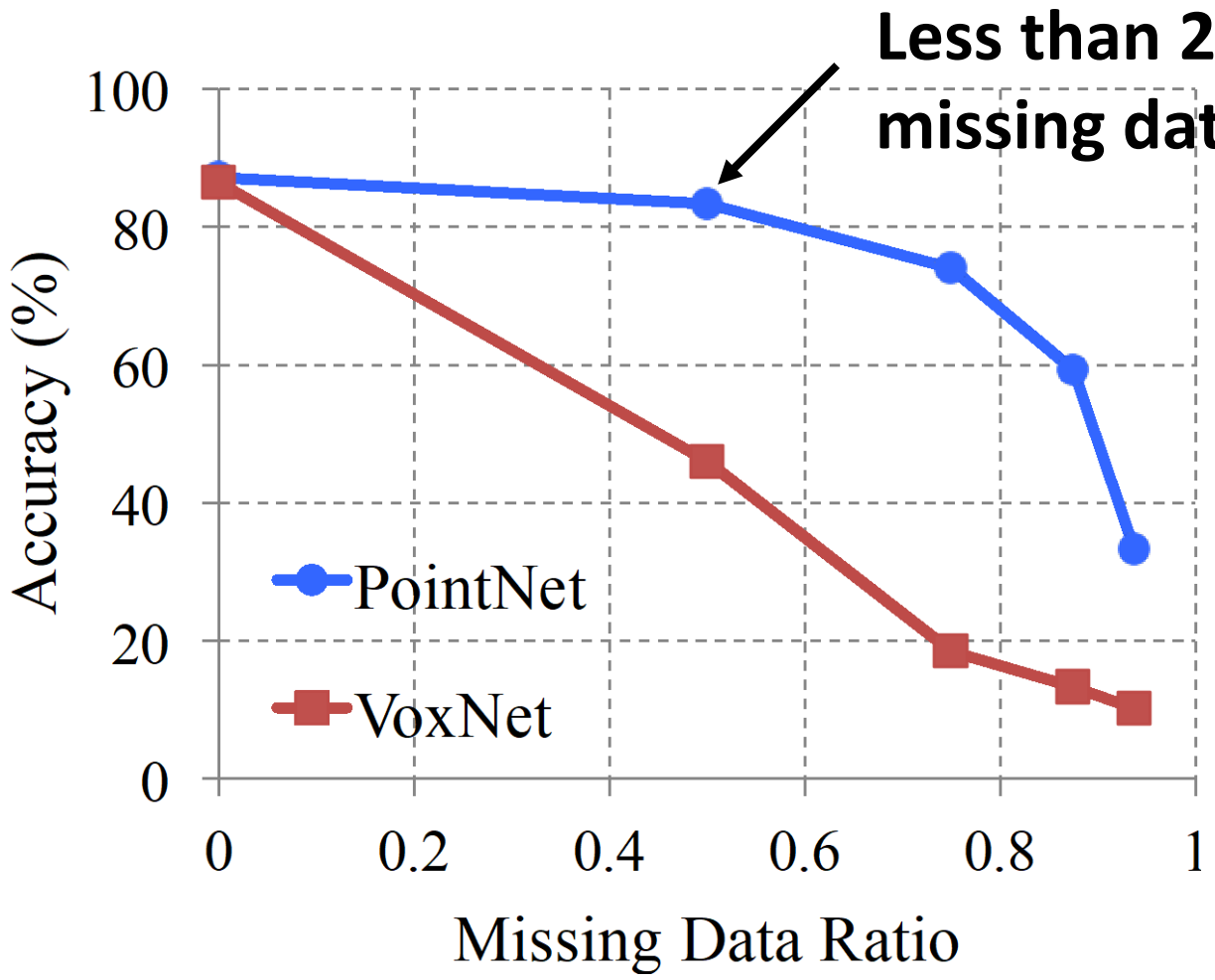
# PointNet is Light-Weight and Fast




**Saves 88% FLOPs**

**A promising architecture for portable devices!**

# PointNet is Robust to Data Corruption



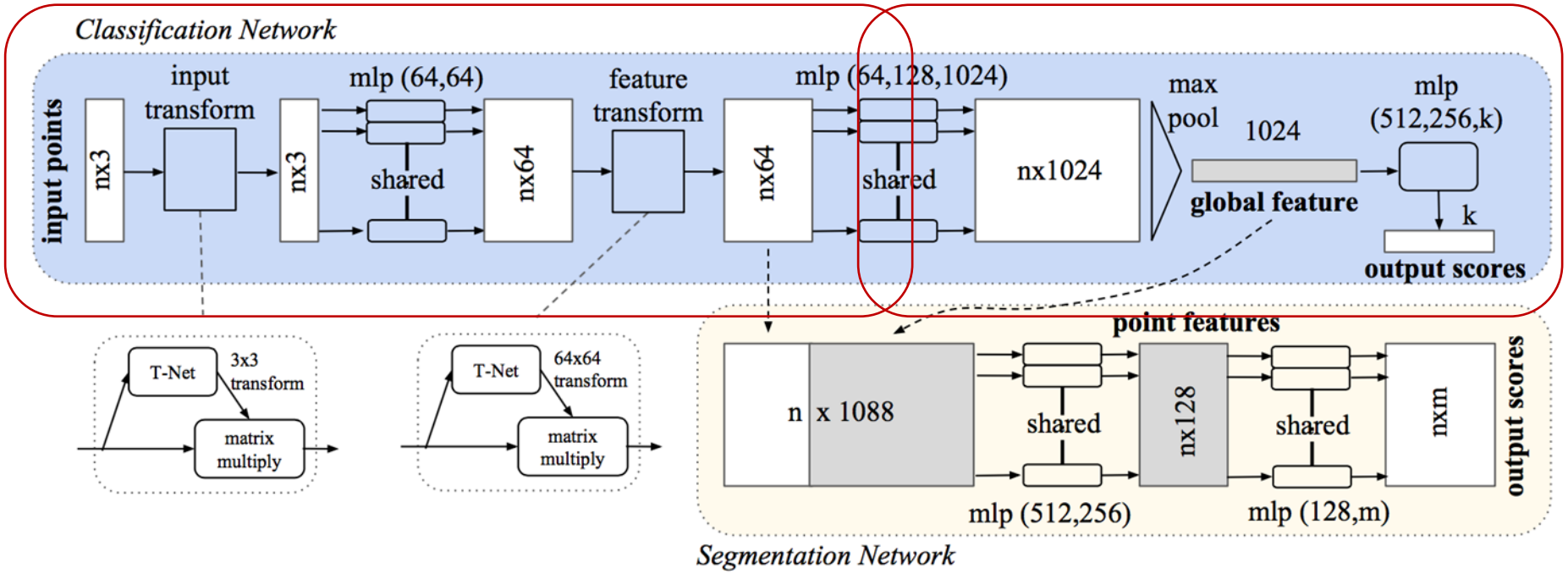

***Why is PointNet so robust to missing data?***

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

# Visualizing Global Point Cloud Features

Original Shape

# Learning Interesting Points

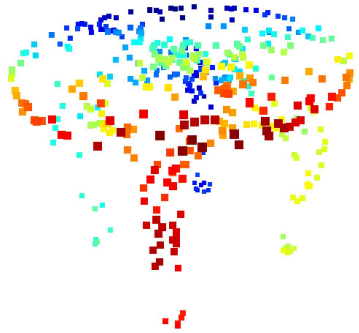


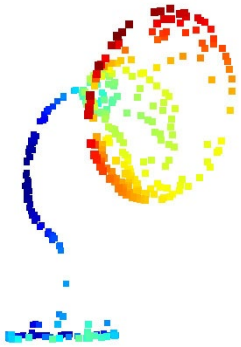

Pointnet learns optimization criteria, which in turn pick interesting points

# Visualizing Global Point Cloud Features

Original Shape

Critical Points



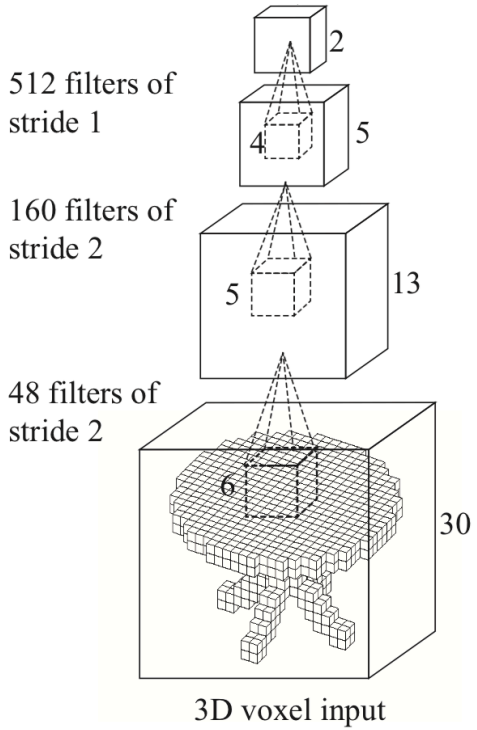


PointNet *learns to pick perceptually interesting points*
A semantic *core-set* …

# From PointNet to PointNet++

# Limitations of PointNet

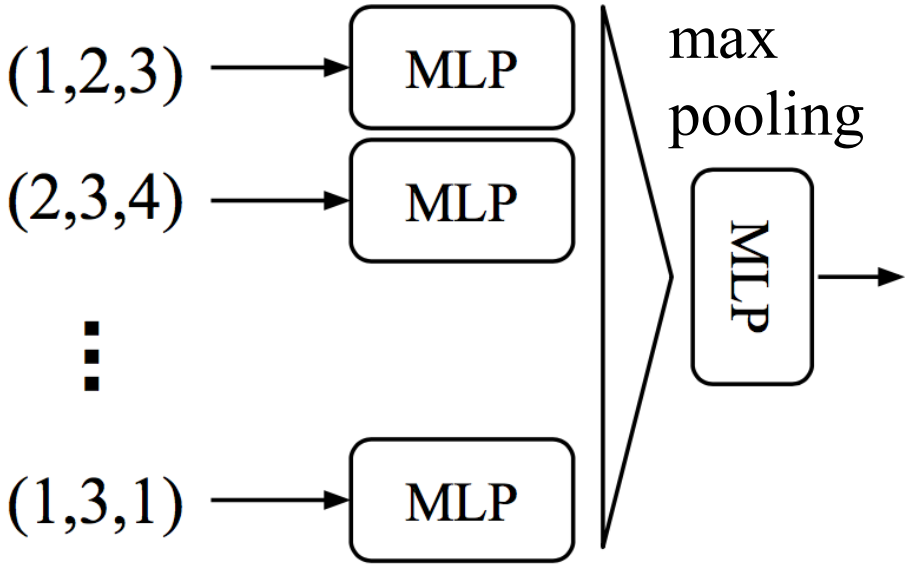
**Hierarchical feature learning**
multiple levels of abstraction




3D CNN [Wu et al.2015]

**v.s.**

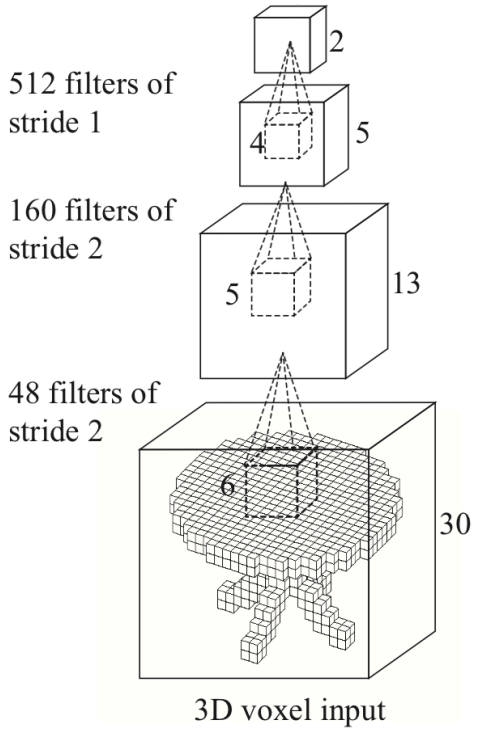
**Global feature learning**
either **one** point, or **all** points




PointNet (vanilla) [Qi et al.2017]

# Limitations of PointNet

**Hierarchical feature learning**
multiple levels of abstraction




3D CNN [Wu et al.2015]

**v.s.**

**Global feature learning**
either **one** point or **all** points

**No local context**

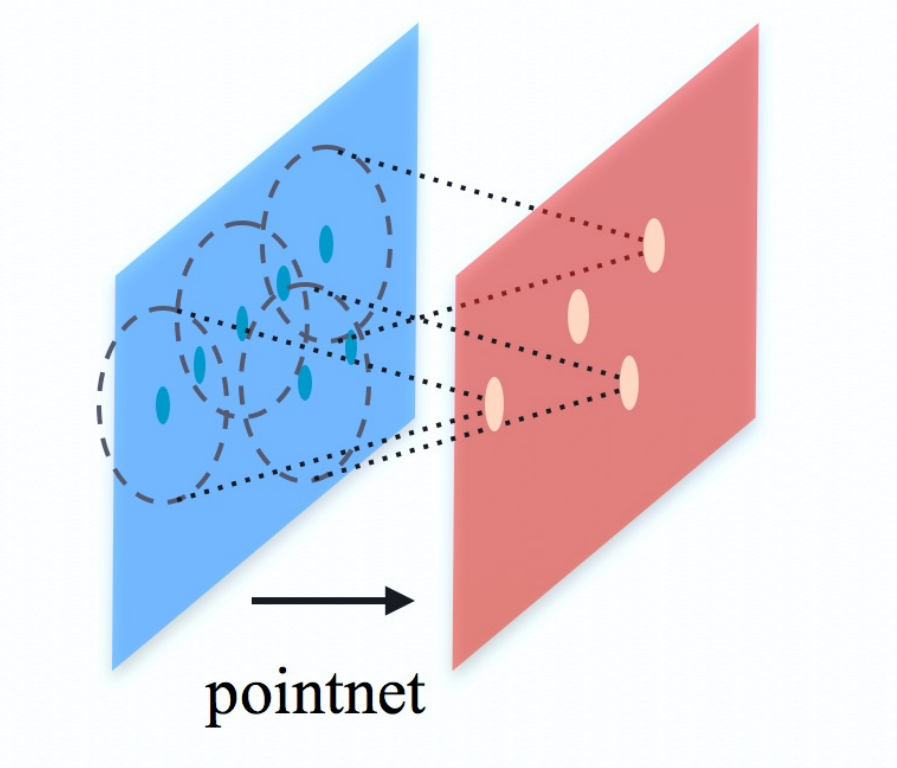
**Limited local invariance**

PointNet (vanilla) [Qi et al.2017]

# PointNet++

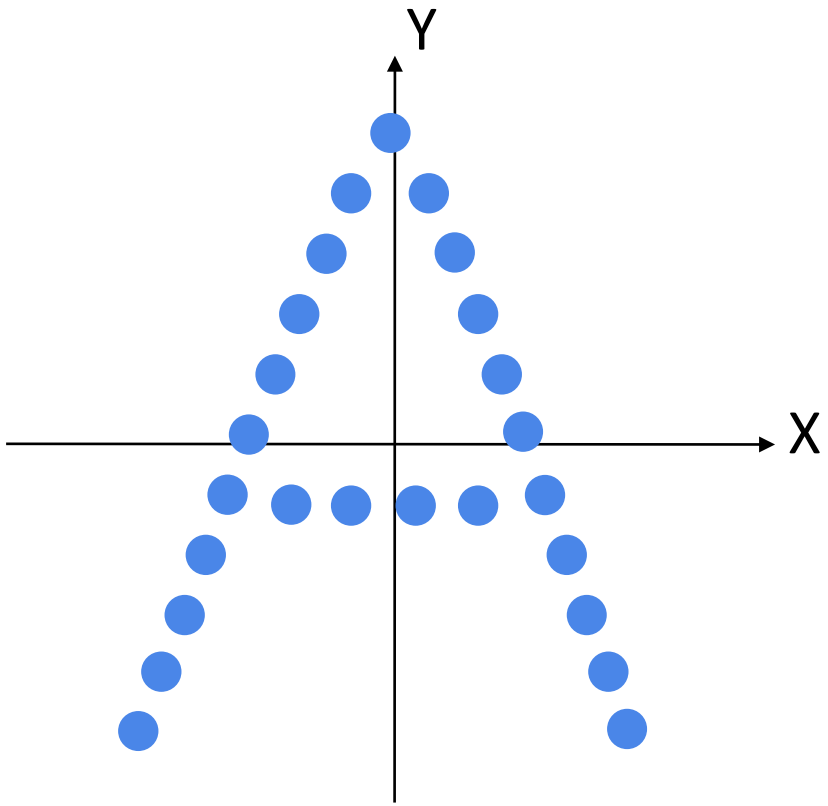
Basic idea: Recursively apply pointnet at local regions.

- ✓ Hierarchical feature learning
- ✓ Local translation invariance
- ✓ Permutation invariance



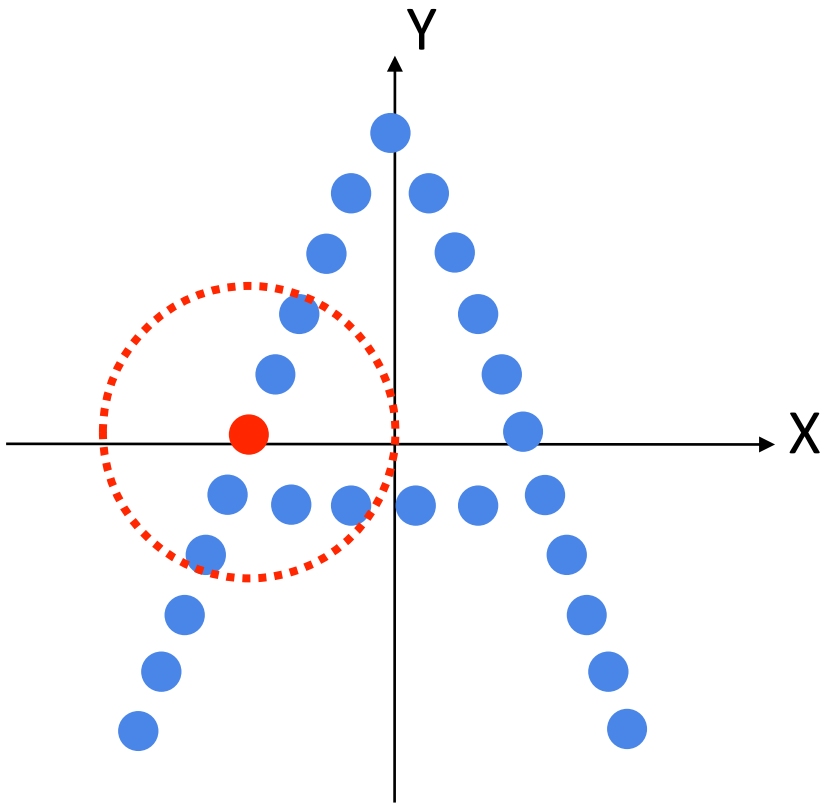

Charles R. Qi, Li Yi, Hao Su, Leonidas Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space (NIPS'17)

# Hierarchical Point Feature Learning



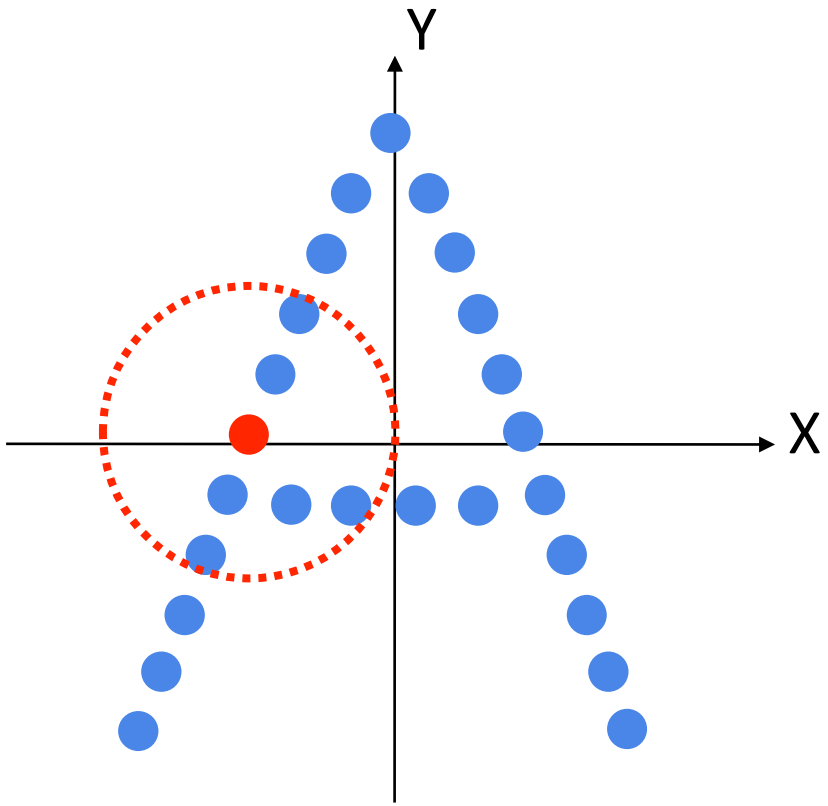

N points in (X,Y)

# Hierarchical Point Feature Learning

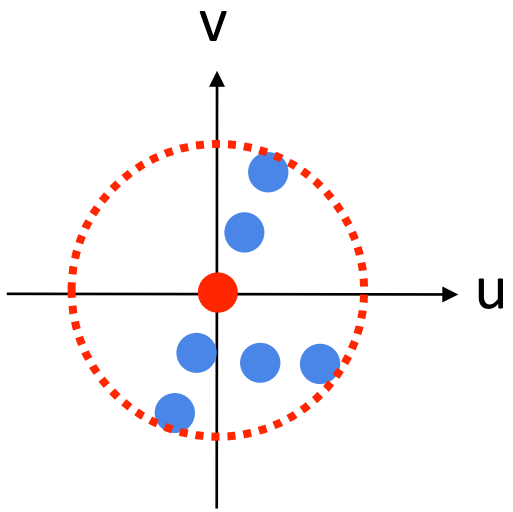



N points in (X,Y)

# Hierarchical Point Feature Learning

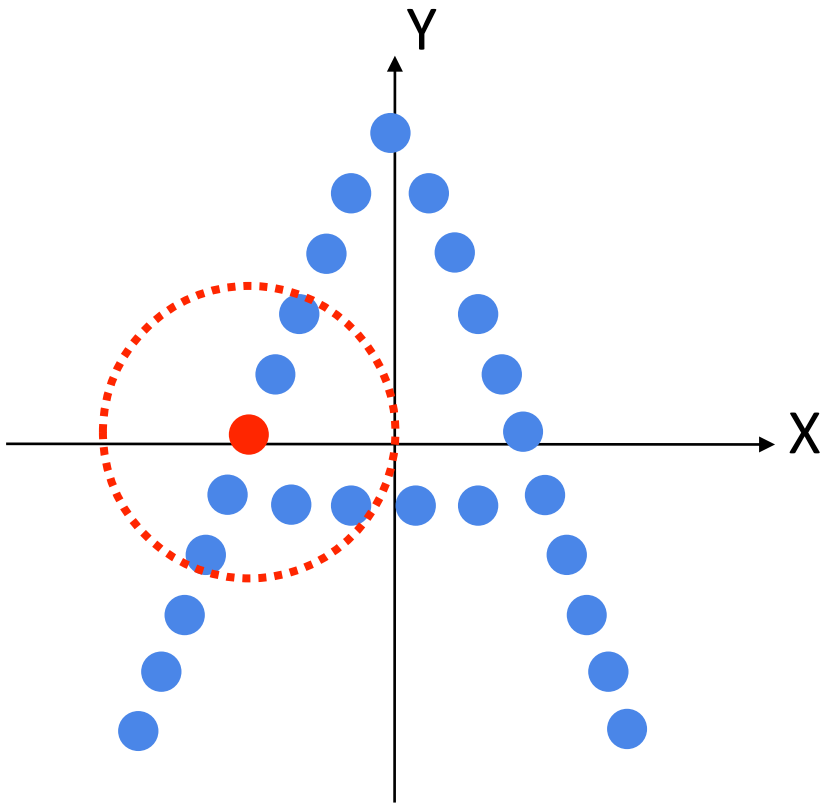



N points in (X,Y)



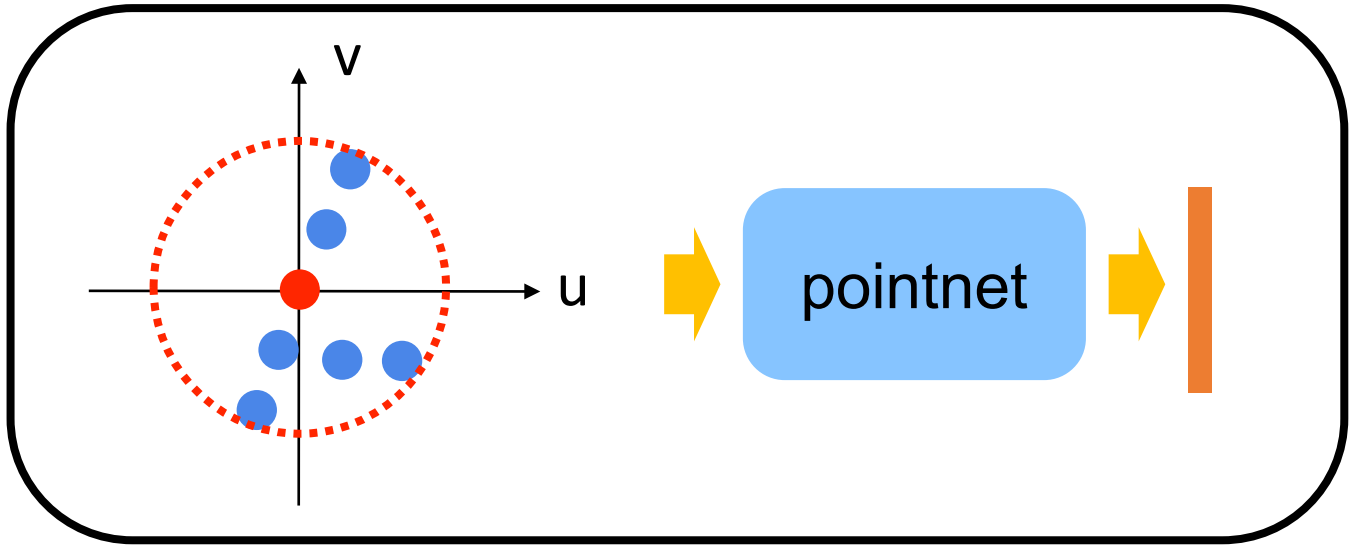

k points in local coordinates (u,v)

# Hierarchical Point Feature Learning



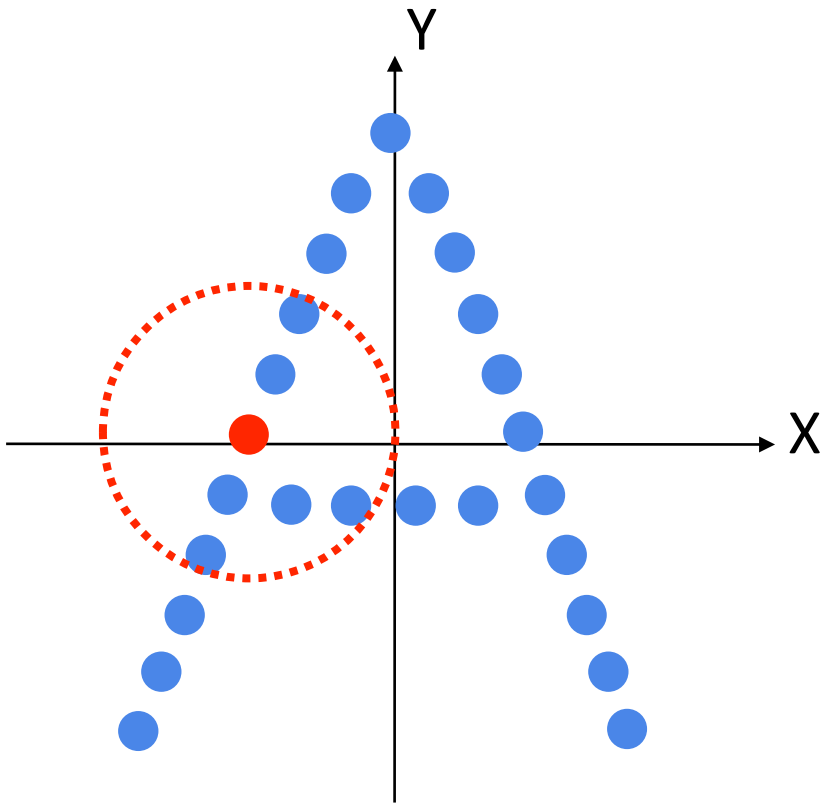

N points in (X,Y)

**Apply pointnet at a local region**

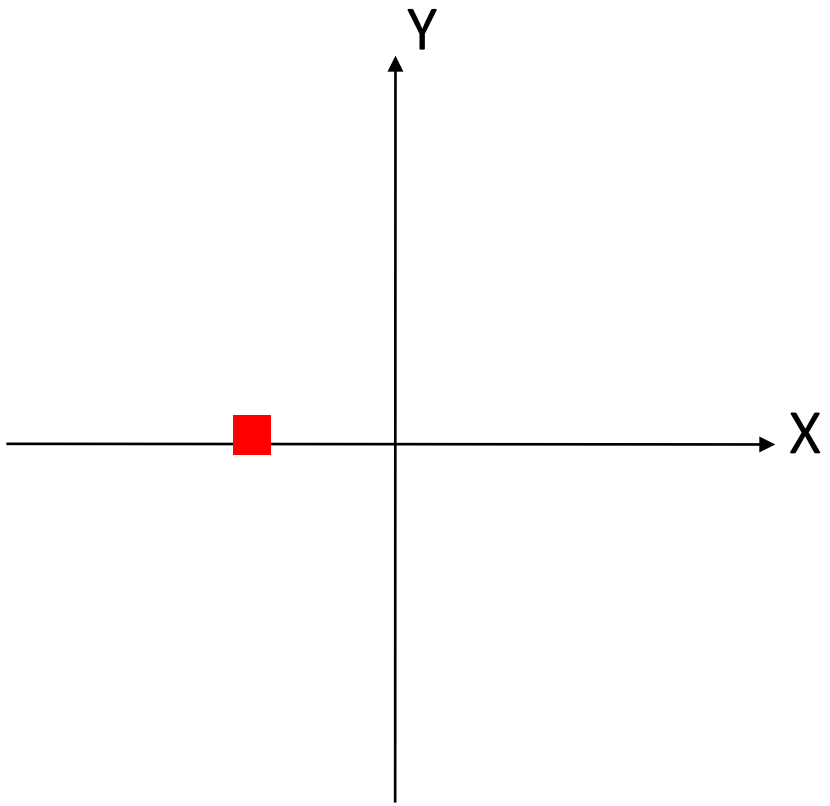



k points in local coordinates (u,v)

# Hierarchical Point Feature Learning




N points in (X,Y)

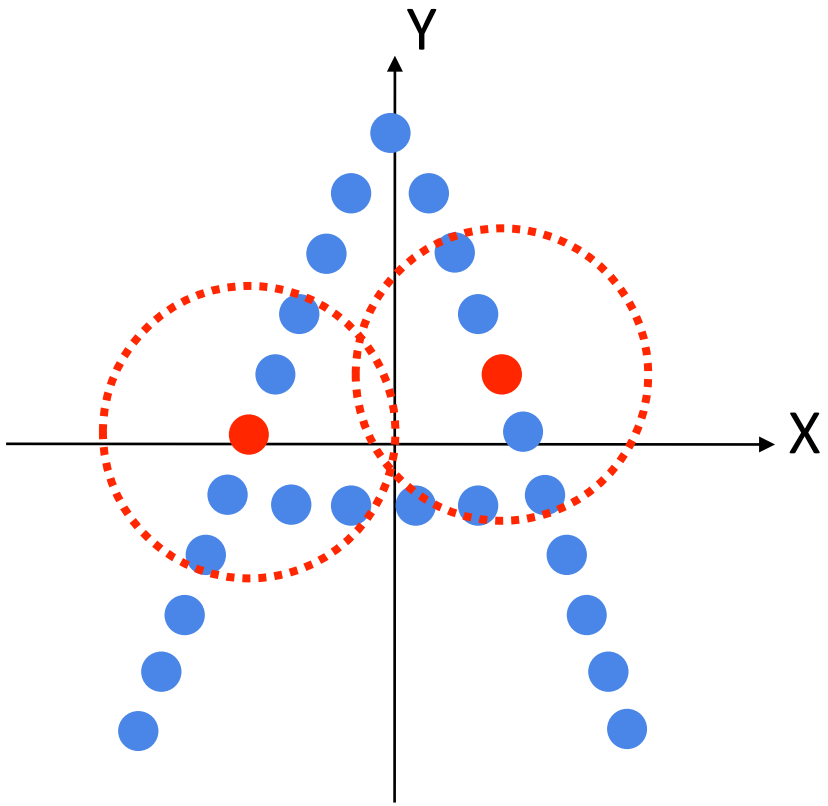



points in (X,Y, **F**)

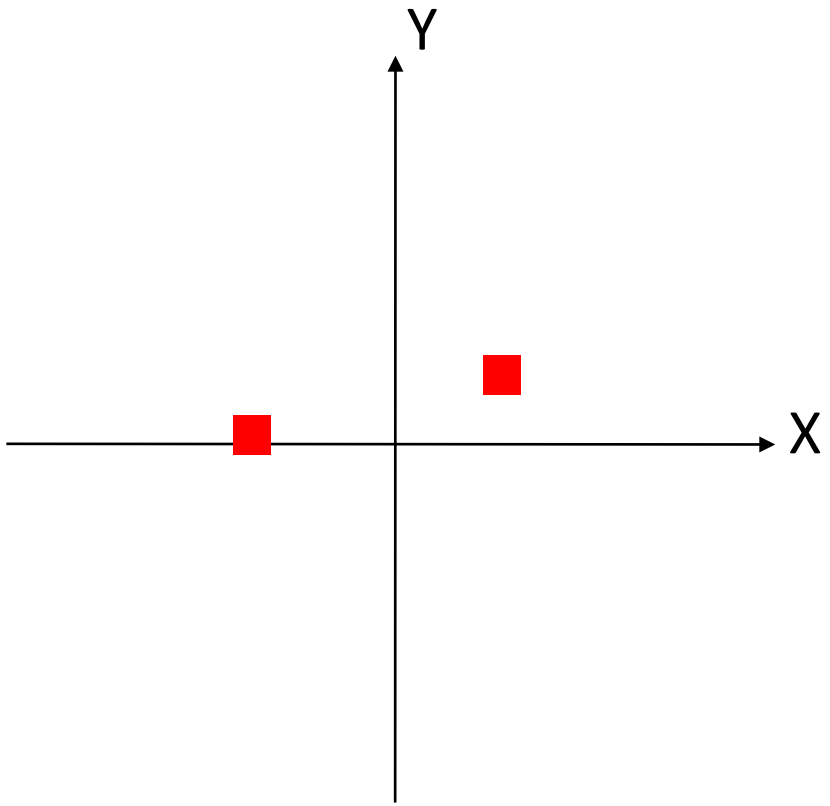
Euclidean space

**high-dim feature space**

# Hierarchical Point Feature Learning

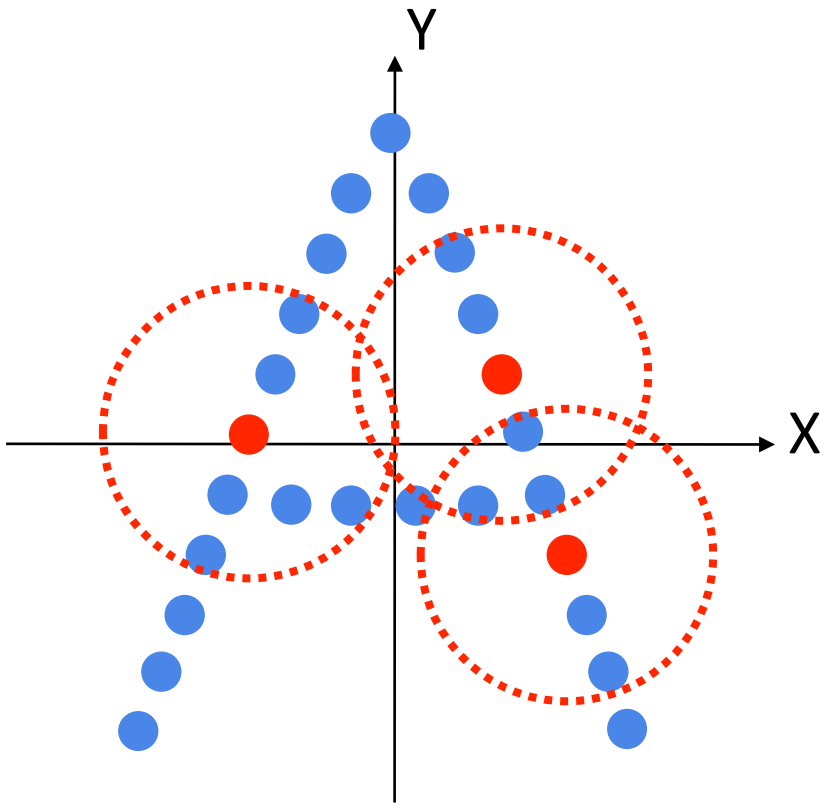



N points in (X,Y)

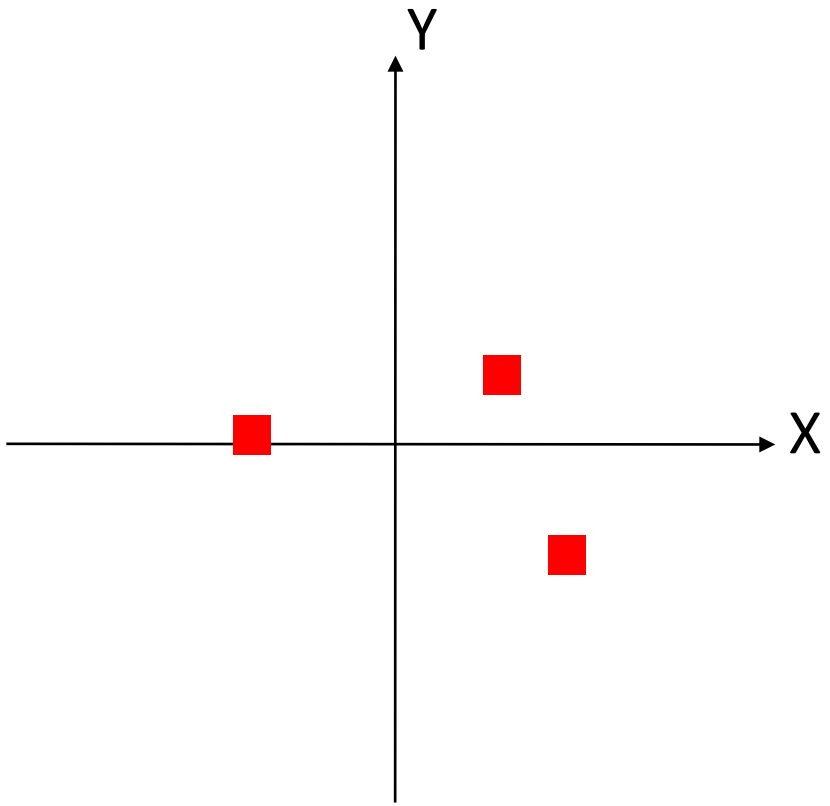



points in (X,Y, **F**)

# Hierarchical Point Feature Learning

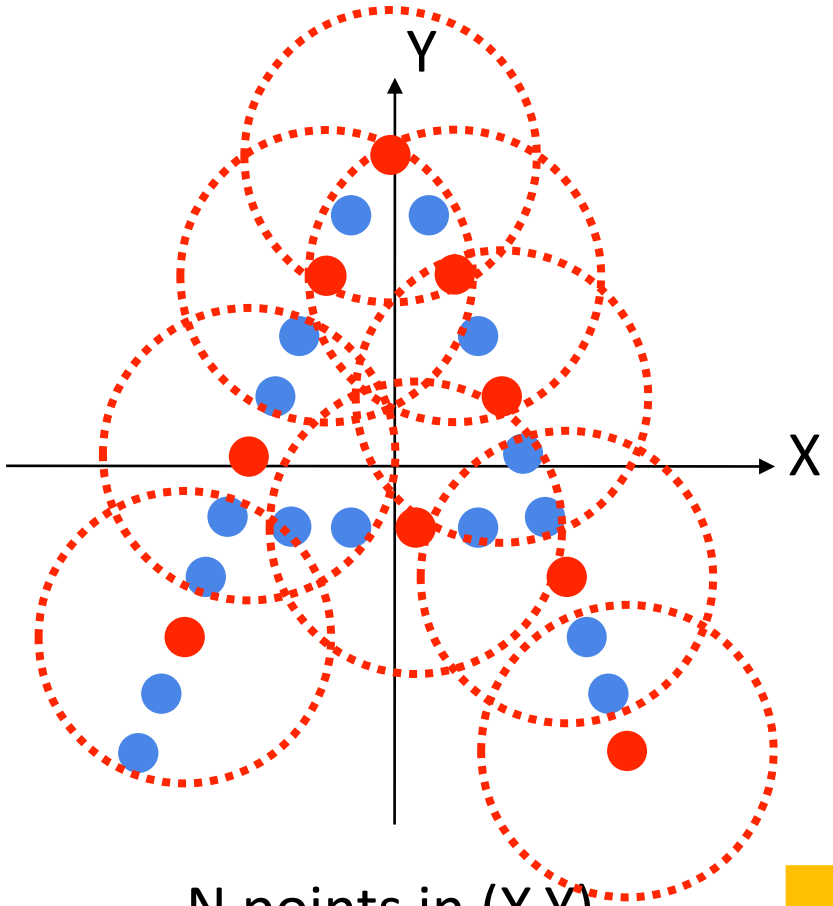



N points in (X,Y)

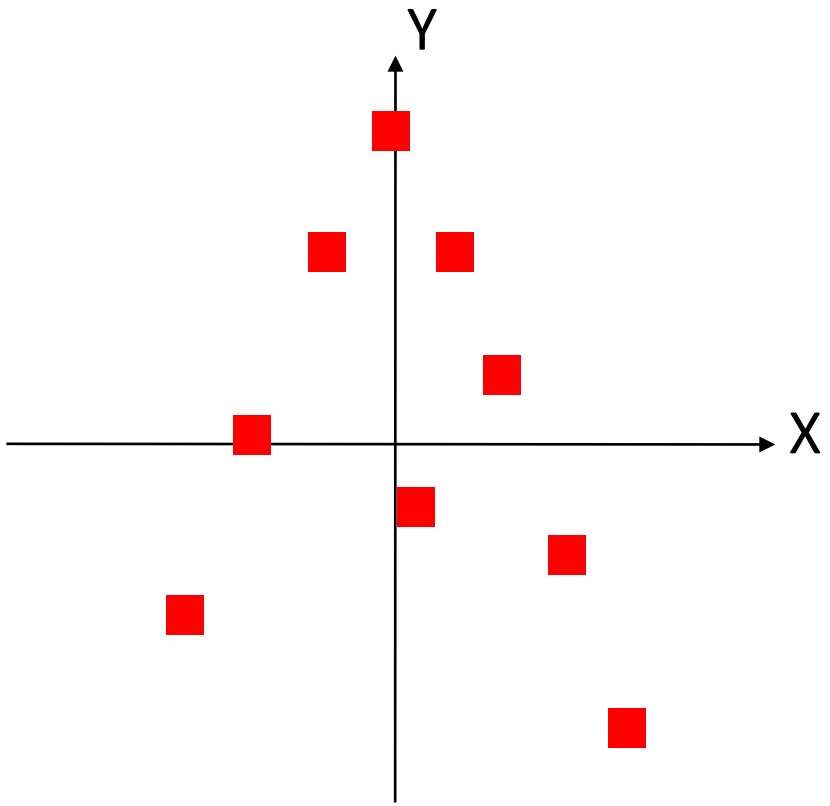



points in (X,Y, **F**)

# Hierarchical Point Feature Learning



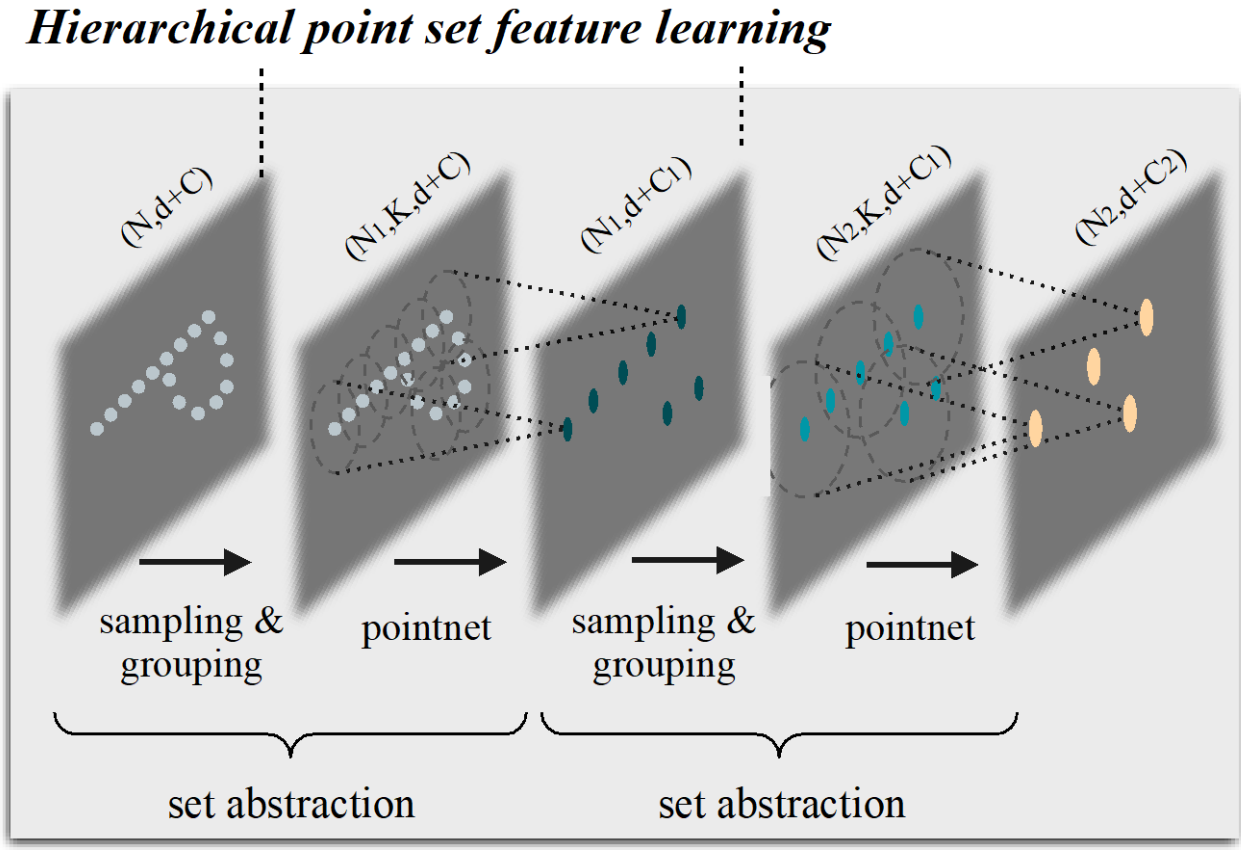

N points in (X,Y)




N1 points in (X,Y, **F**)

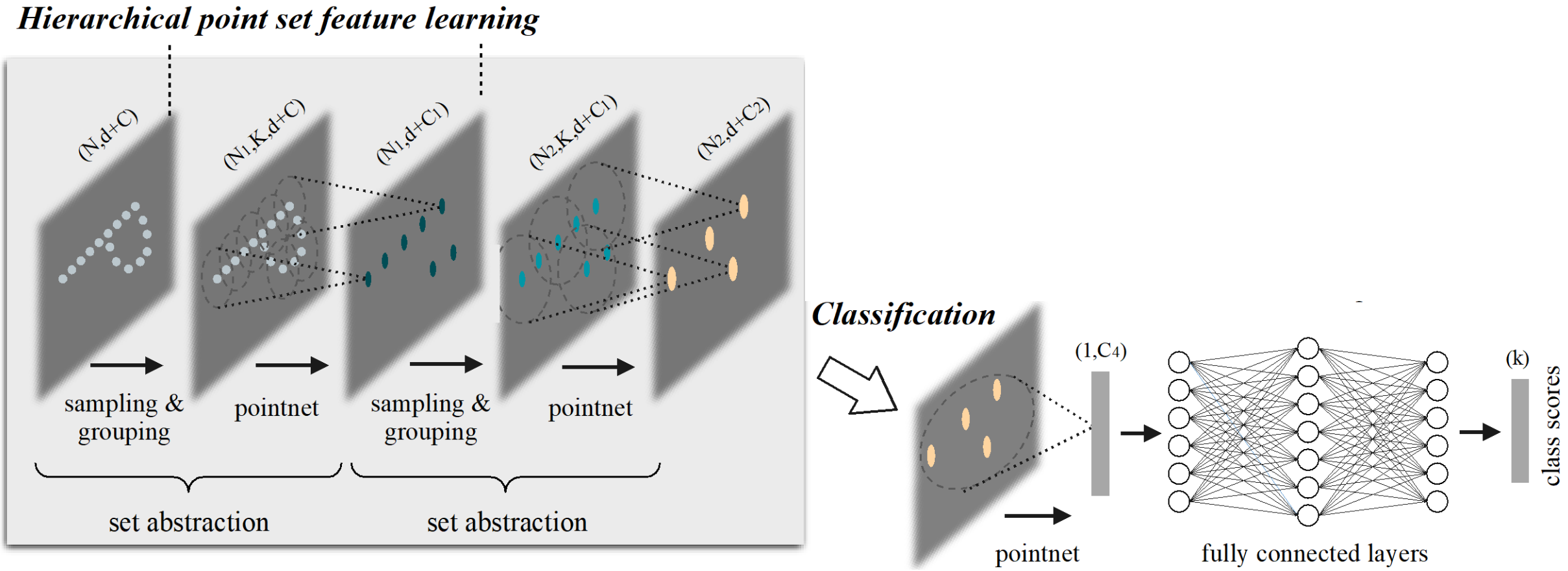
**Set Abstraction:** farthest point sampling + grouping + pointnet

# PointNet++ for Classification and Segmentation

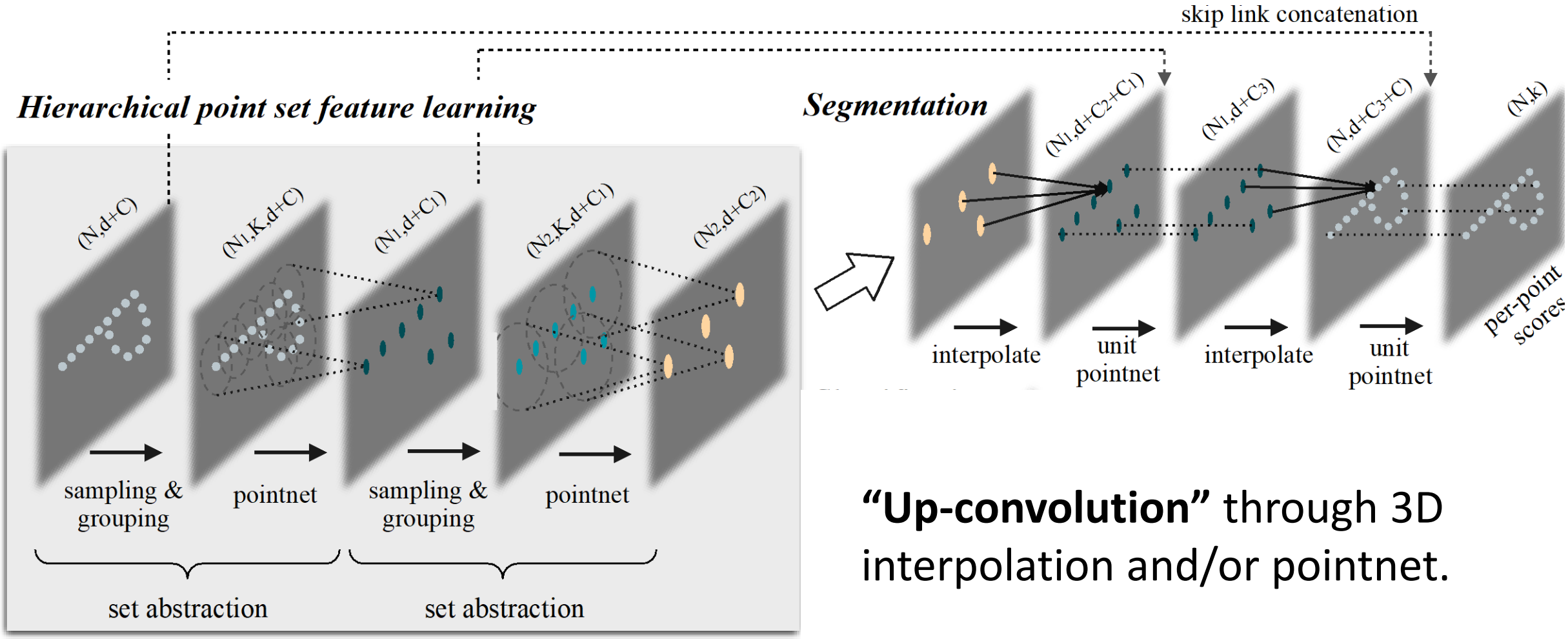



Caveat: Shouldn't feature dimensions from the lower layers affect connectivity at the higher layers?

# PointNet++ for Classification and Segmentation



# PointNet++ for Classification and Segmentation

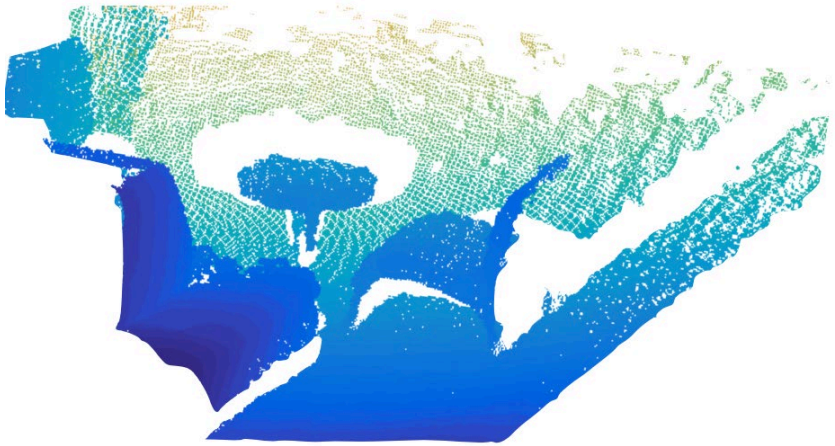



**"Up-convolution"** through 3D interpolation and/or pointnet.

# Non-uniform Sampling Density in Point Clouds

Density variation is a common issue in 3D point cloud processing

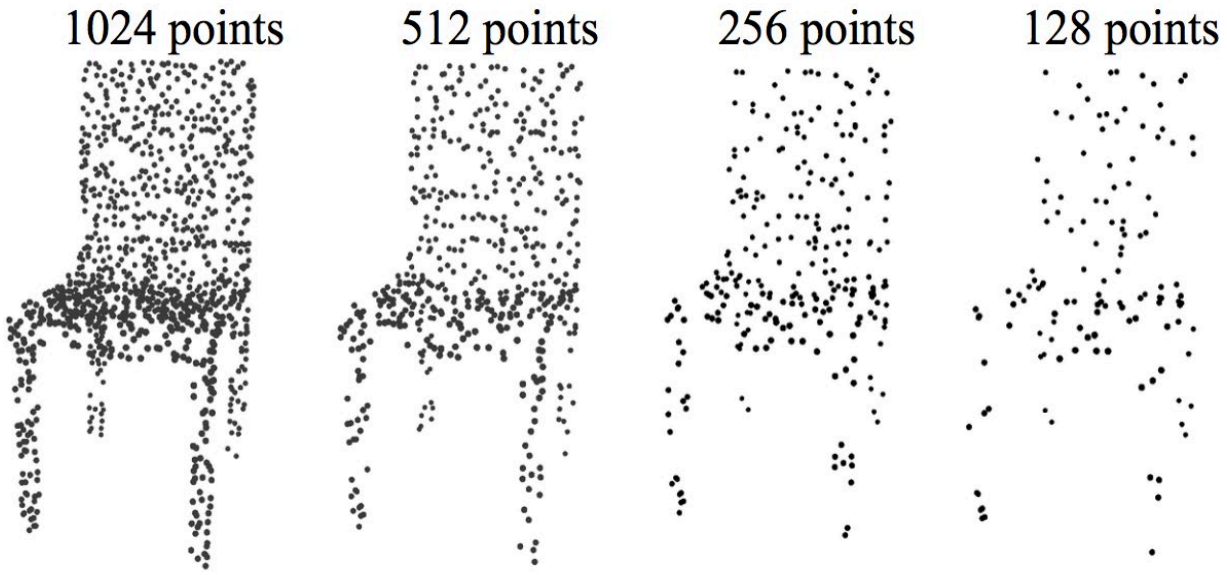
- perspective effect, radial density variation, motion etc.

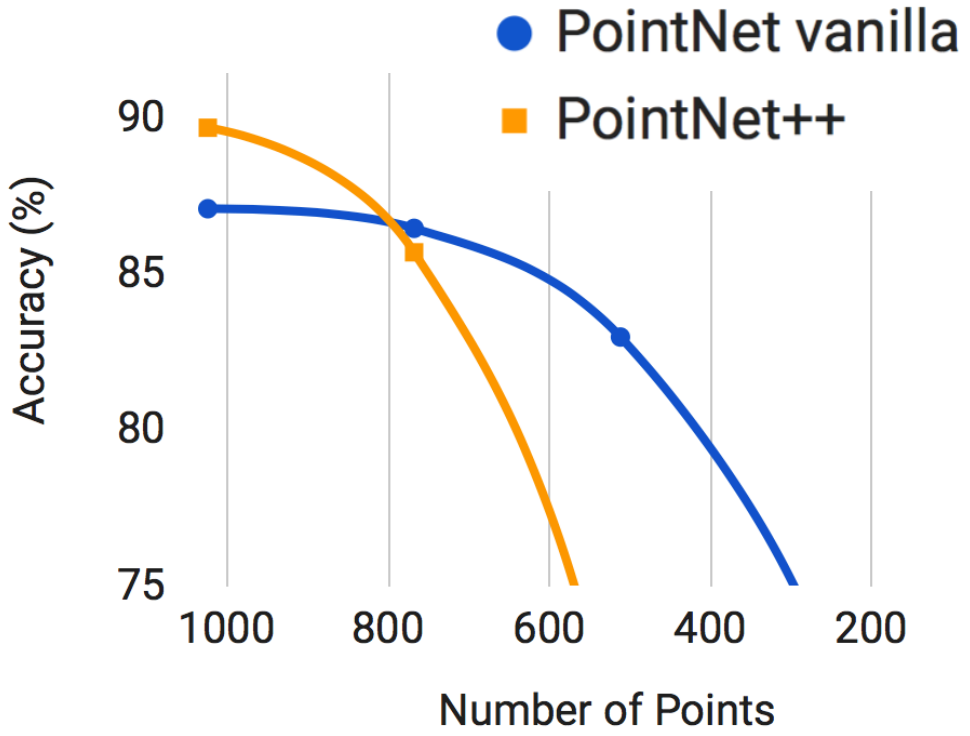


**Challenge for local feature learning!**

Network may learn sampling pattern – not underlying geometry

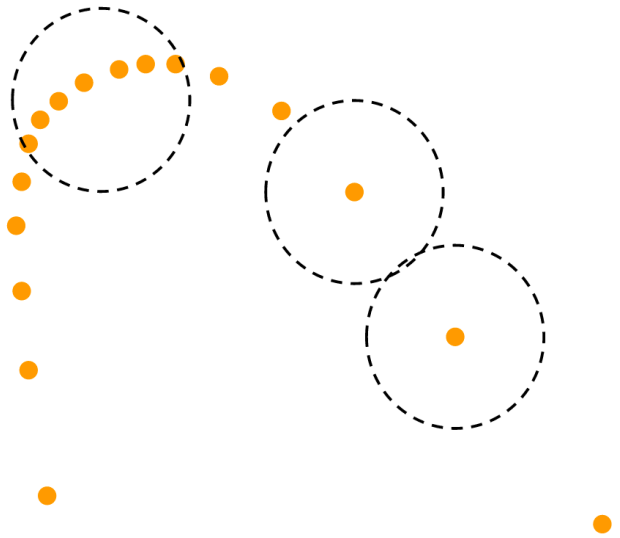
# Density Variation Affects Hierarchy

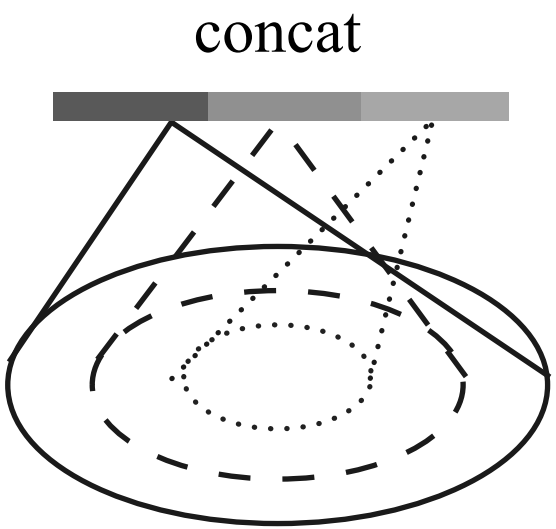



**Small kernels suffer from varying densities!**

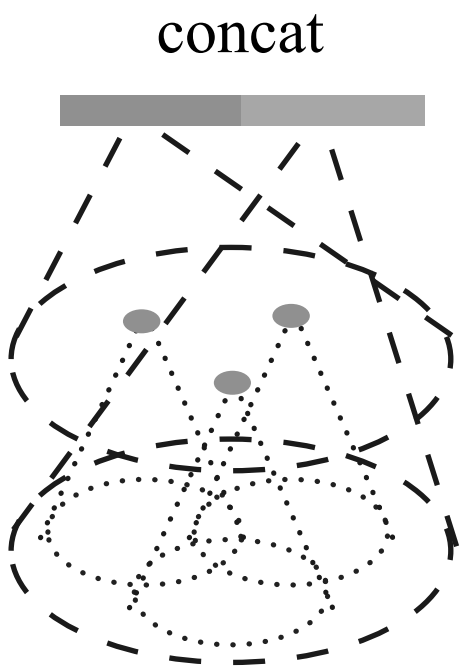
# Robust Learning Under Varying Sampling Density






(a)

Multi-scale grouping (MSG)

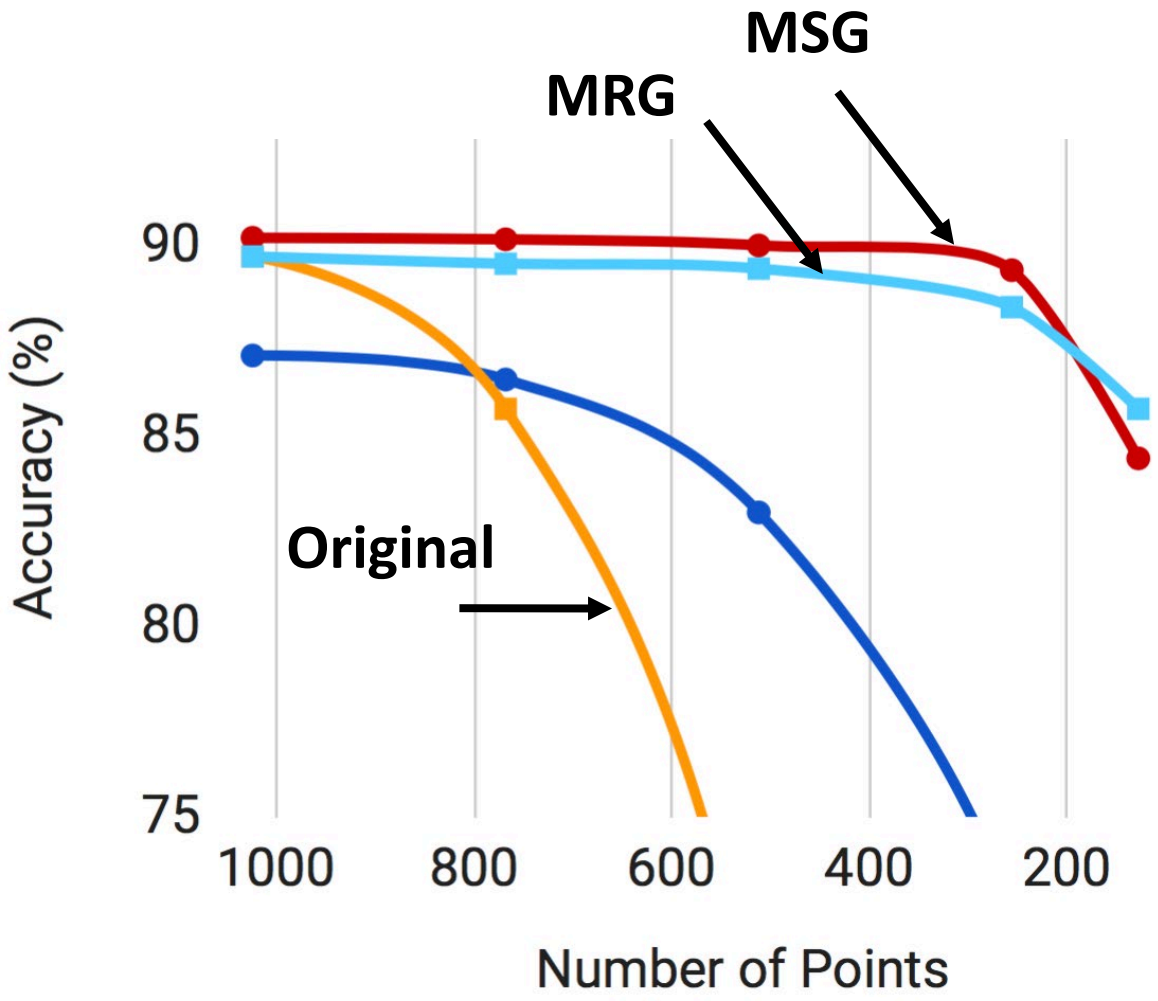



(b)

Multi-res grouping (MRG)

*During Training: input point dropout with random dropout ratio*

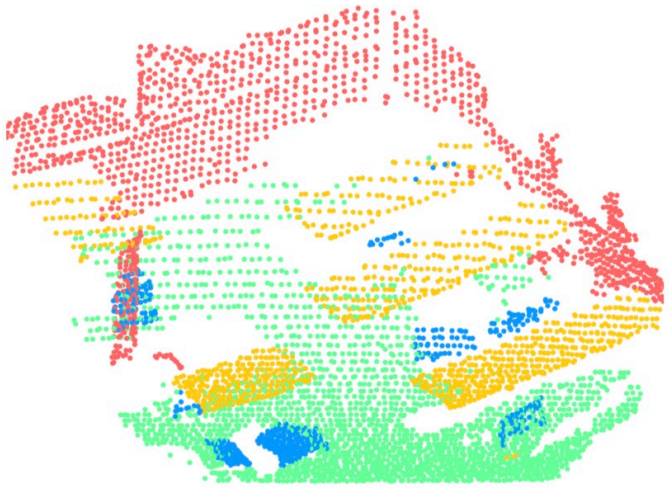
# Robust Learning Under Varying Sampling Density

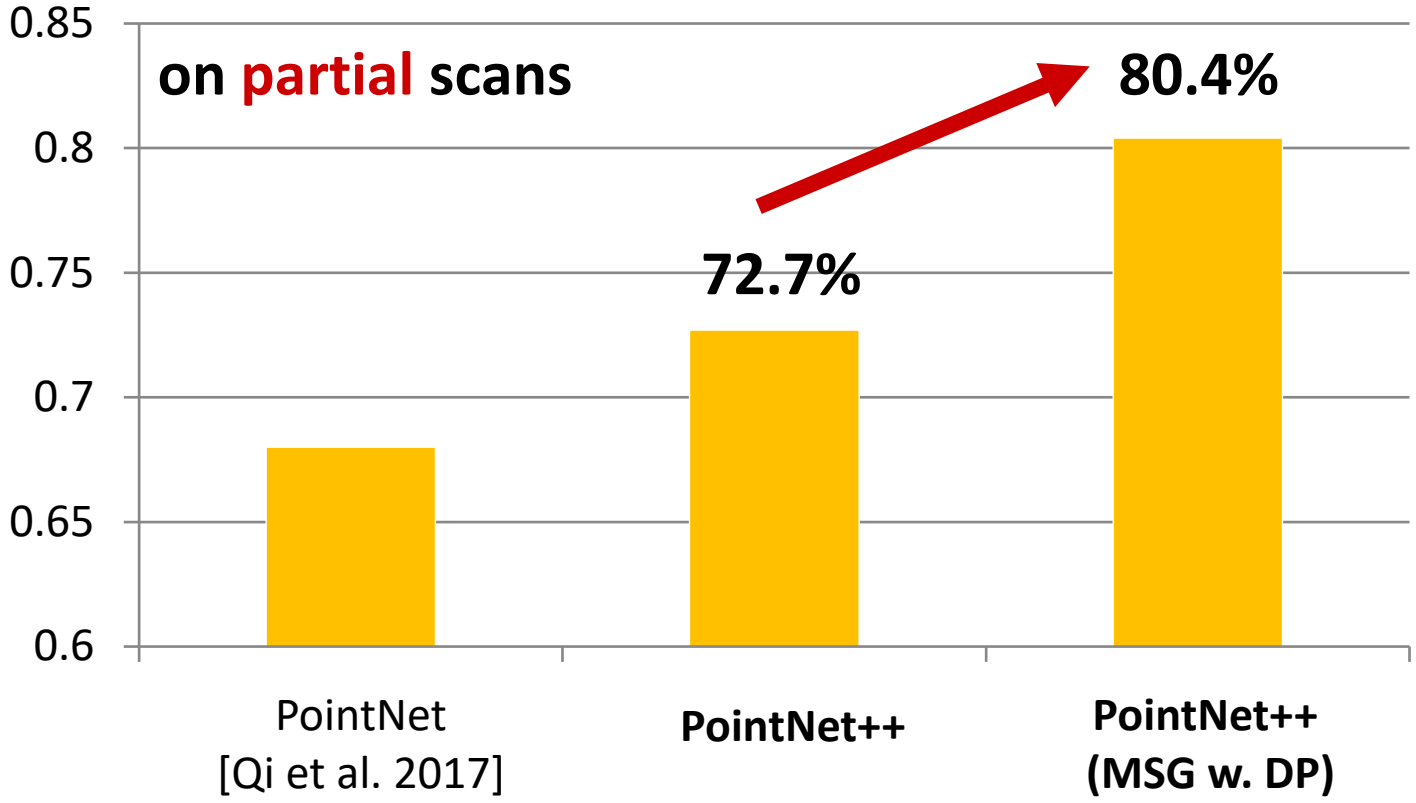



- PointNet vanilla
- PointNet++
- PointNet++ (MSG w. DP)
- PointNet++ (MRG w. DP)

# PointNet++ Results: Scene Parsing

**Robust layers for non-uniform densities (MSG) help a lot.**

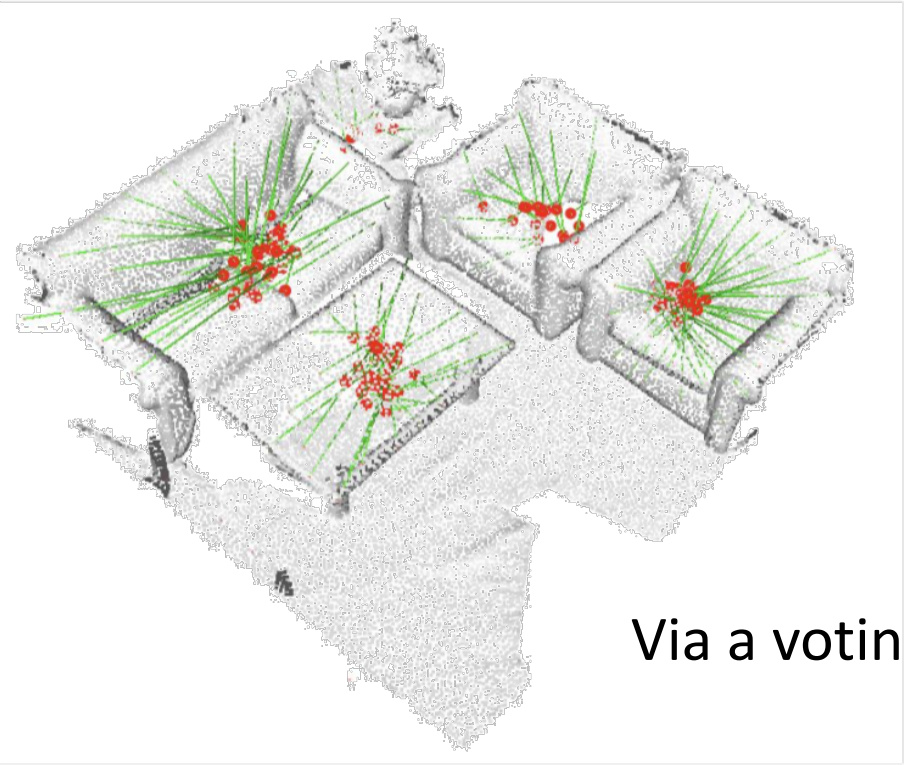


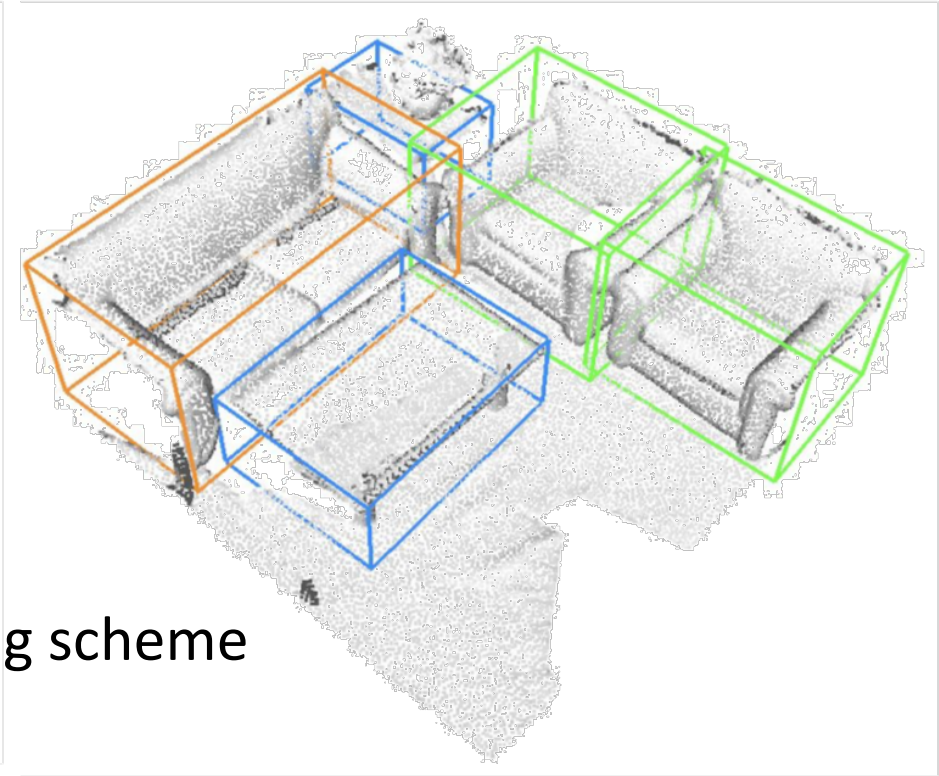



*dataset: ScanNet; metric: per-point semantic classification accuracy (%)*

# Object Detection in Point Clouds, Indoors

# Point Cloud Object Amodal Bounding Box Detection

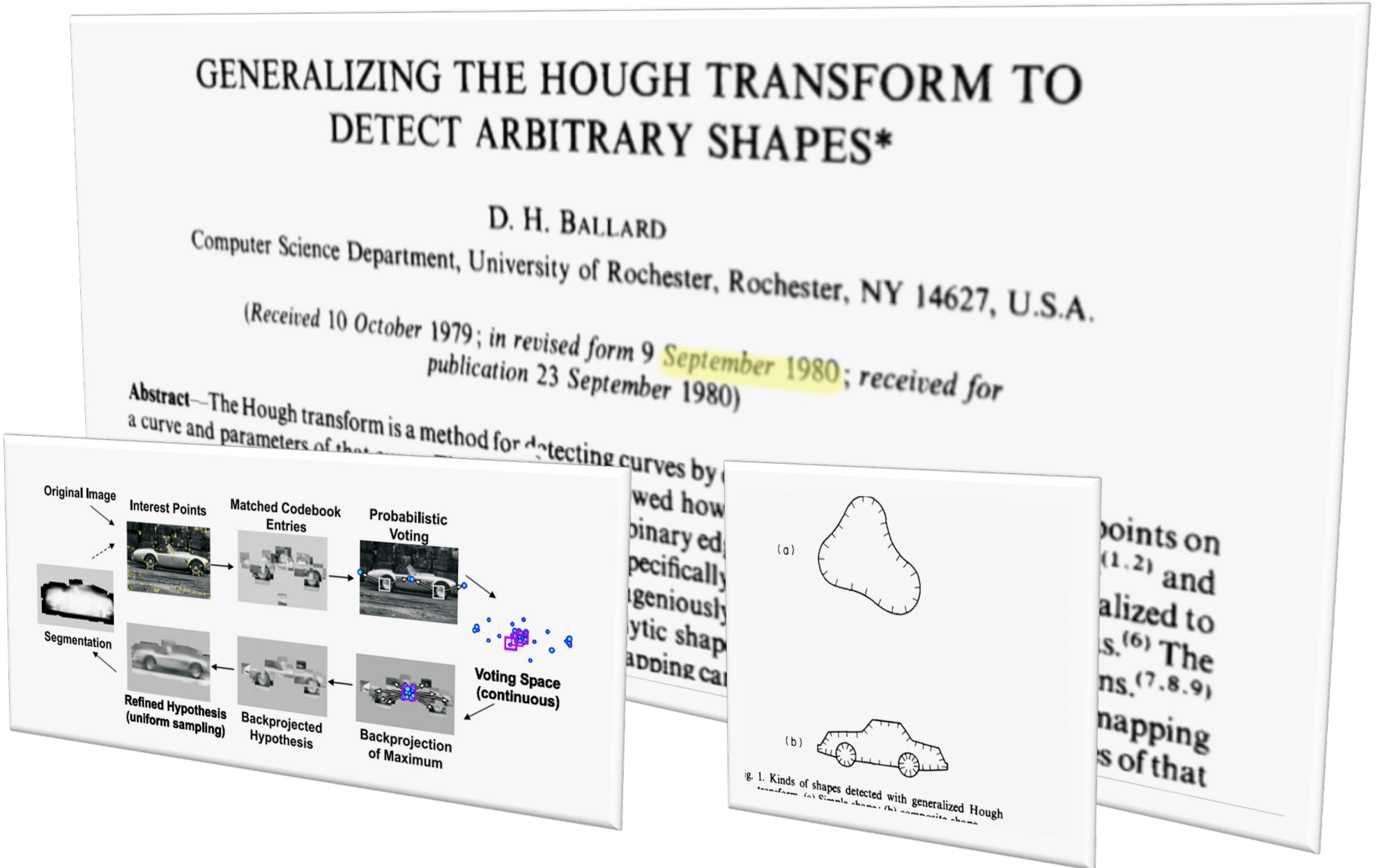




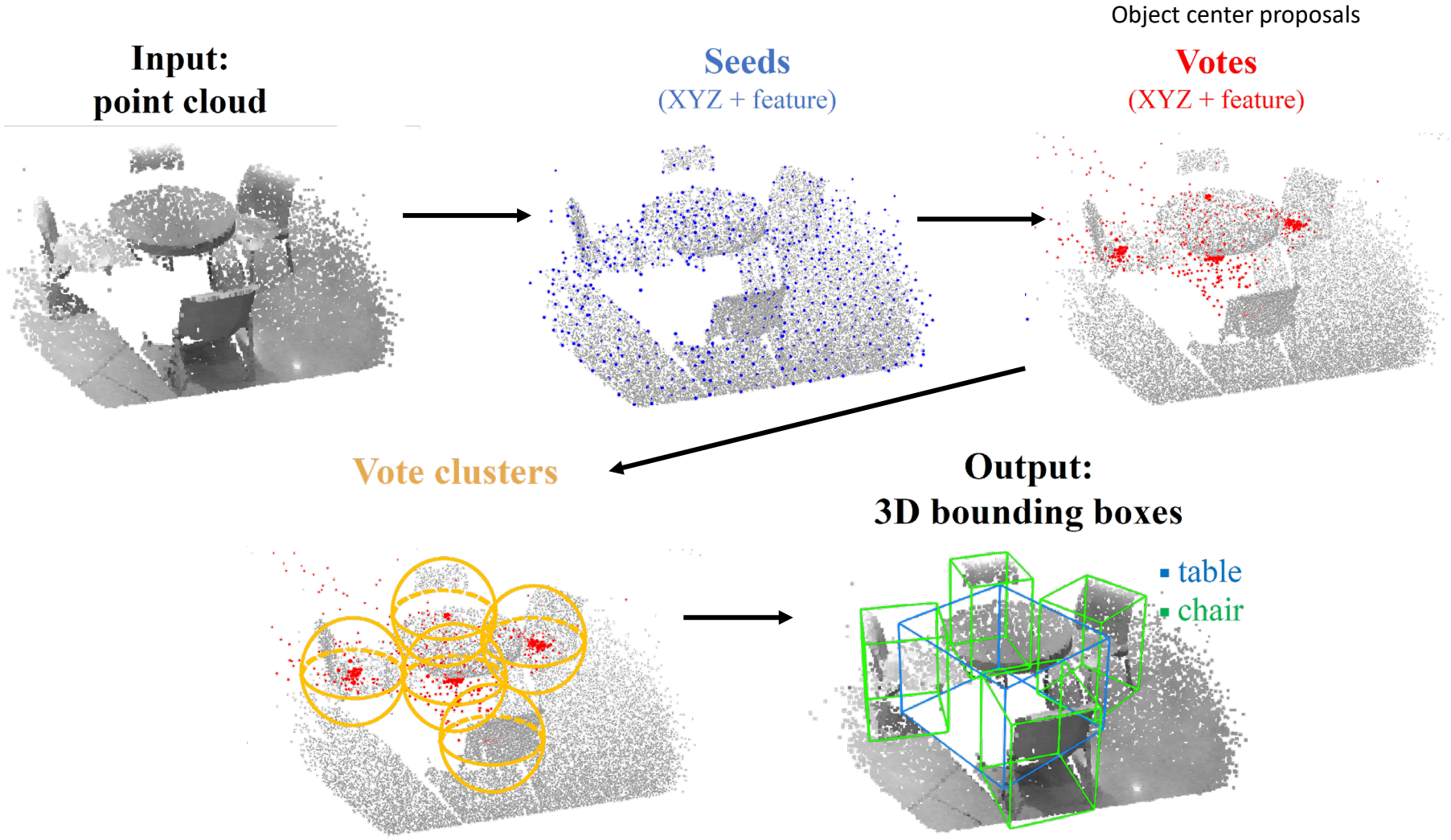
Via a voting scheme

[Charles R. Qi, Or Litany, Kaiming He, Leonidas J. Guibas.
Deep Hough Voting for 3D Object Detection in Point Clouds. ICCV '19]

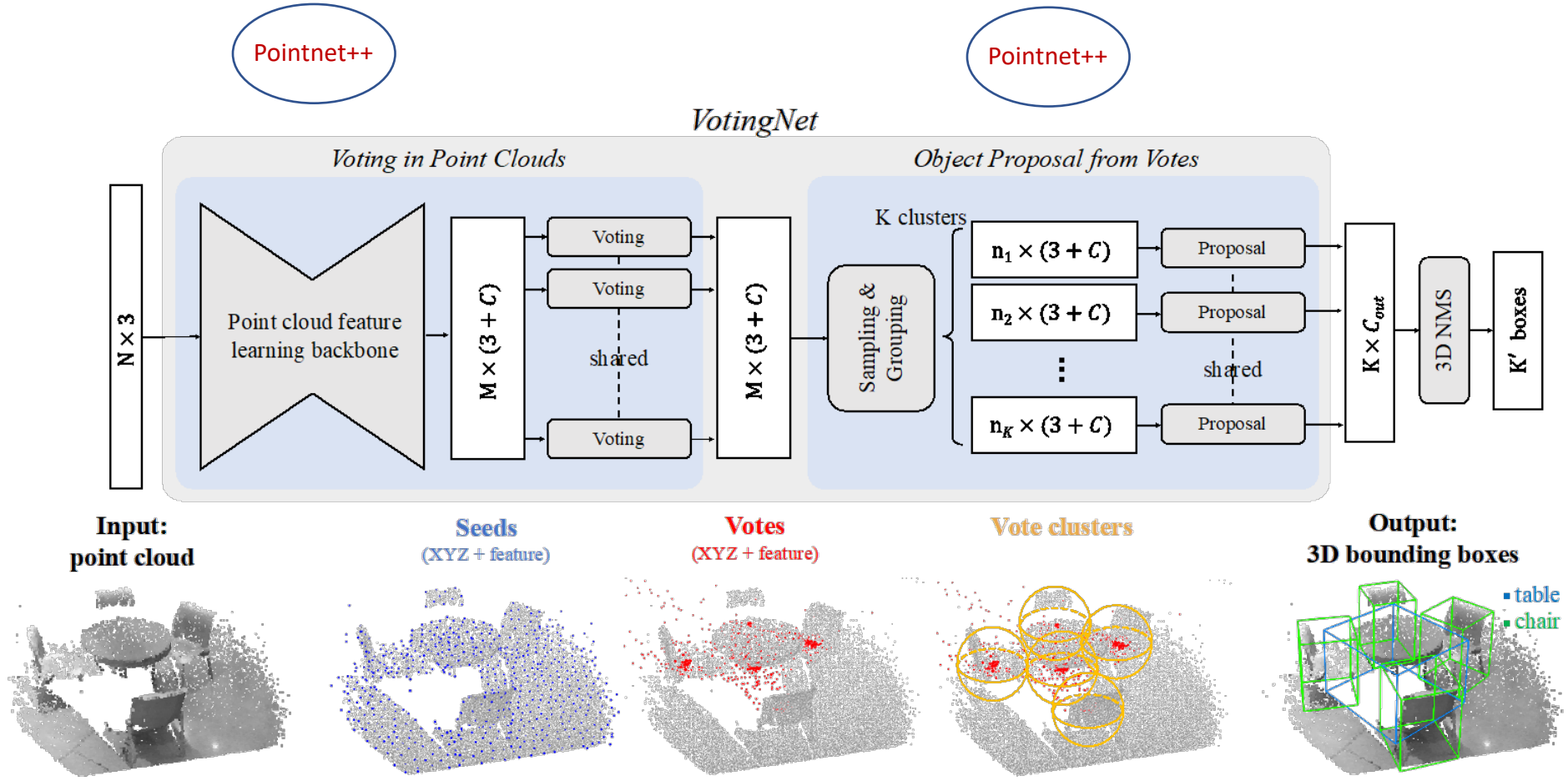
# Generalized Hough Transform



# Deep Hough Voting – A Two-Stage Approach



# VoteNet – A Two-Stage Approach



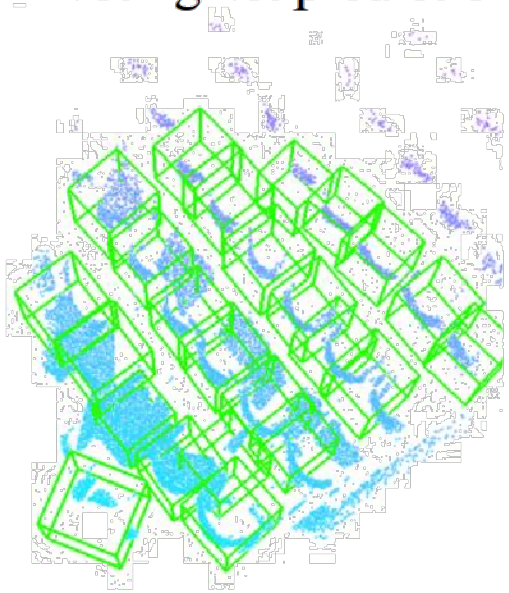

A capsule network in disguise …

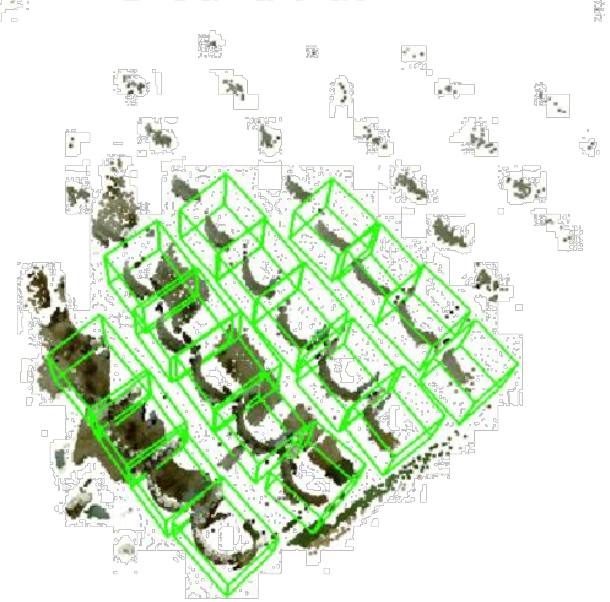
# Results on SUN RGB-D

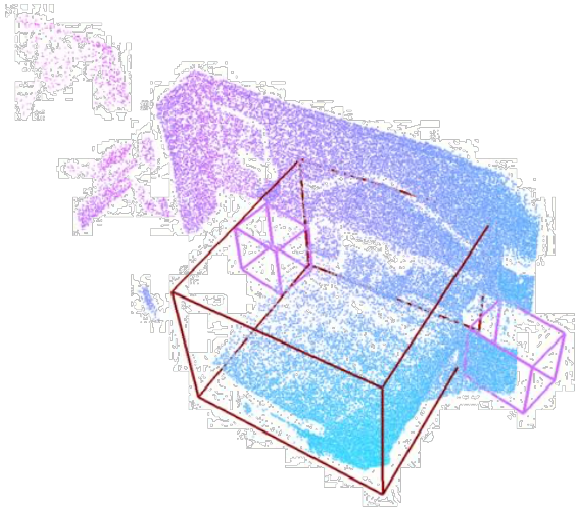
Image of the scene

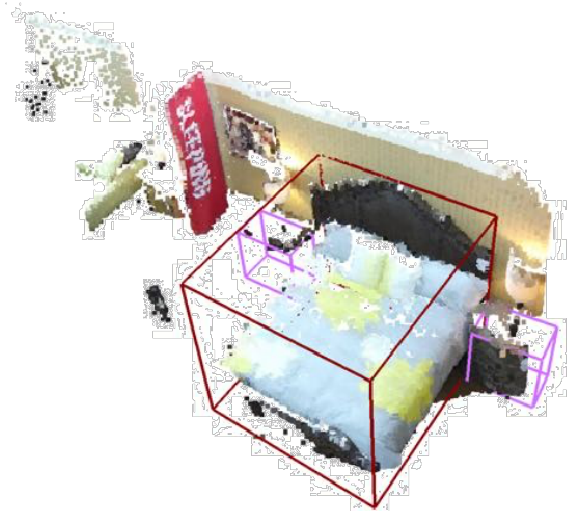
VotingNet prediction

Ground truth





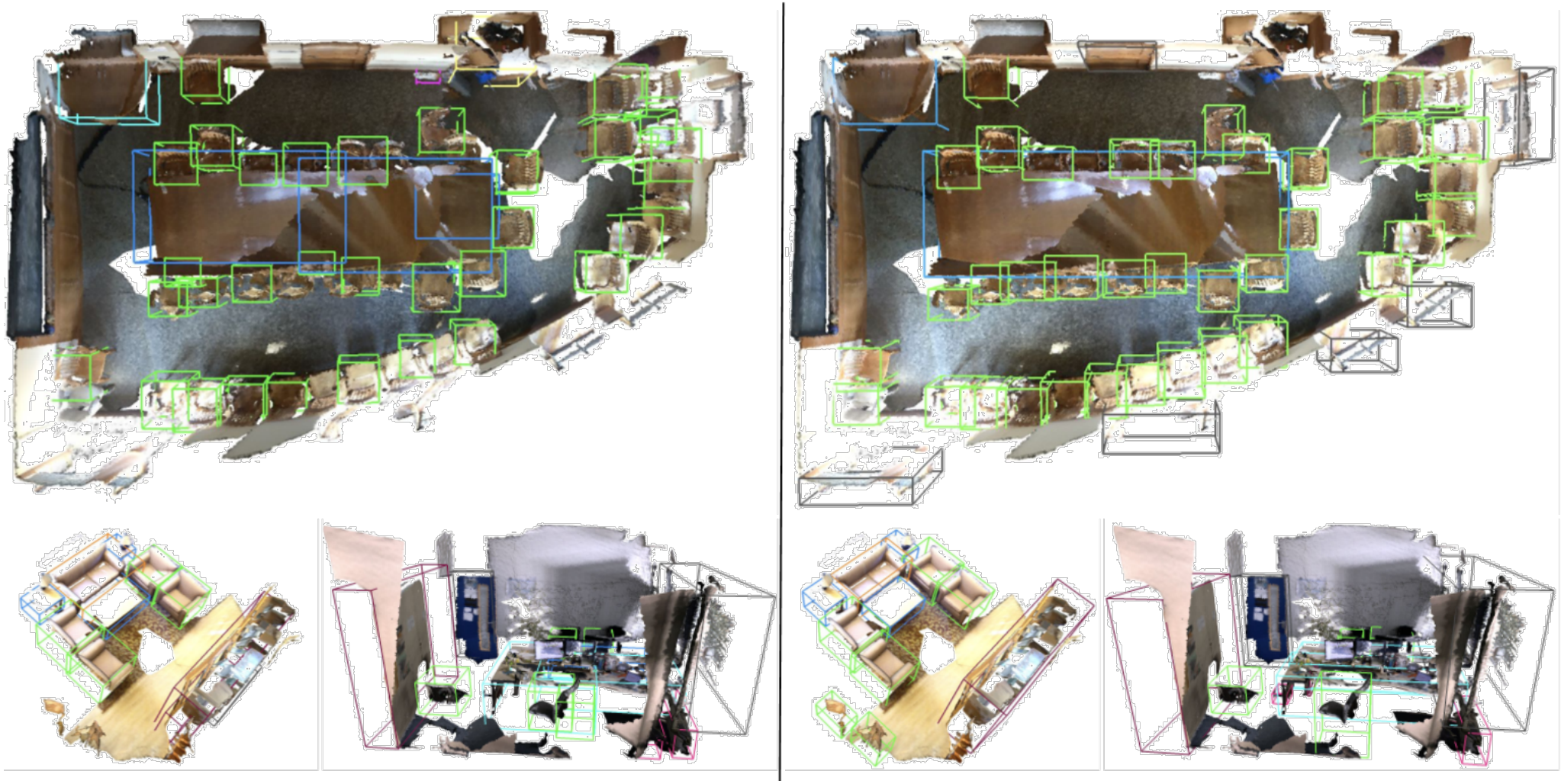





# Results on ScanNet

VotingNet prediction

Ground truth



# Quantitative Results

Deep sliding shapes
Clouds of oriented gradients

Frustum pointnet

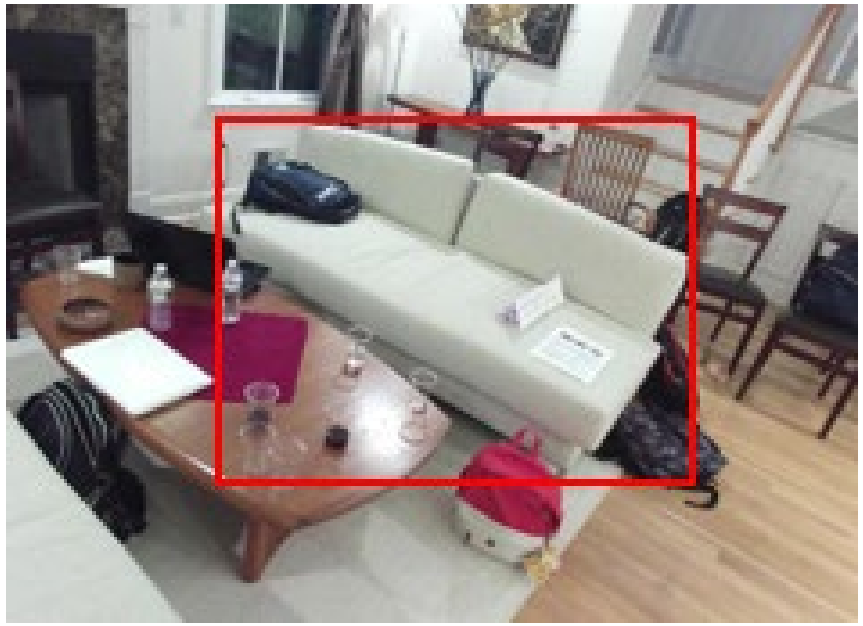
SUN RGB-D

| | Input | bathtub | bed | bookshelf | chair | desk | dresser | nightstand | sofa | table | toilet | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DSS [37] | Geo + RGB | 44.2 | 78.8 | 11.9 | 61.2 | 20.5 | 6.4 | 15.4 | 53.5 | 50.3 | 78.9 | 42.1 |
| COG [33] | Geo + RGB | 58.3 | 63.7 | 31.8 | 62.2 | **45.2** | 15.5 | 27.4 | 51.0 | **51.3** | 70.1 | 47.6 |
| 2D-driven [17] | Geo + RGB | 43.5 | 64.5 | 31.4 | 48.3 | 27.9 | 25.9 | 41.9 | 50.4 | 37.0 | 80.4 | 45.1 |
| F-PointNet [30] | Geo + RGB | 43.3 | 81.1 | **33.3** | 64.2 | 24.7 | **32.0** | 58.1 | 61.1 | 51.1 | **90.9** | 54.0 |
| VotingNet (ours) | Geo only | **74.4** | **83.0** | 28.8 | **75.3** | 22.0 | 29.8 | **62.2** | **64.0** | 47.3 | 90.1 | **57.7** |

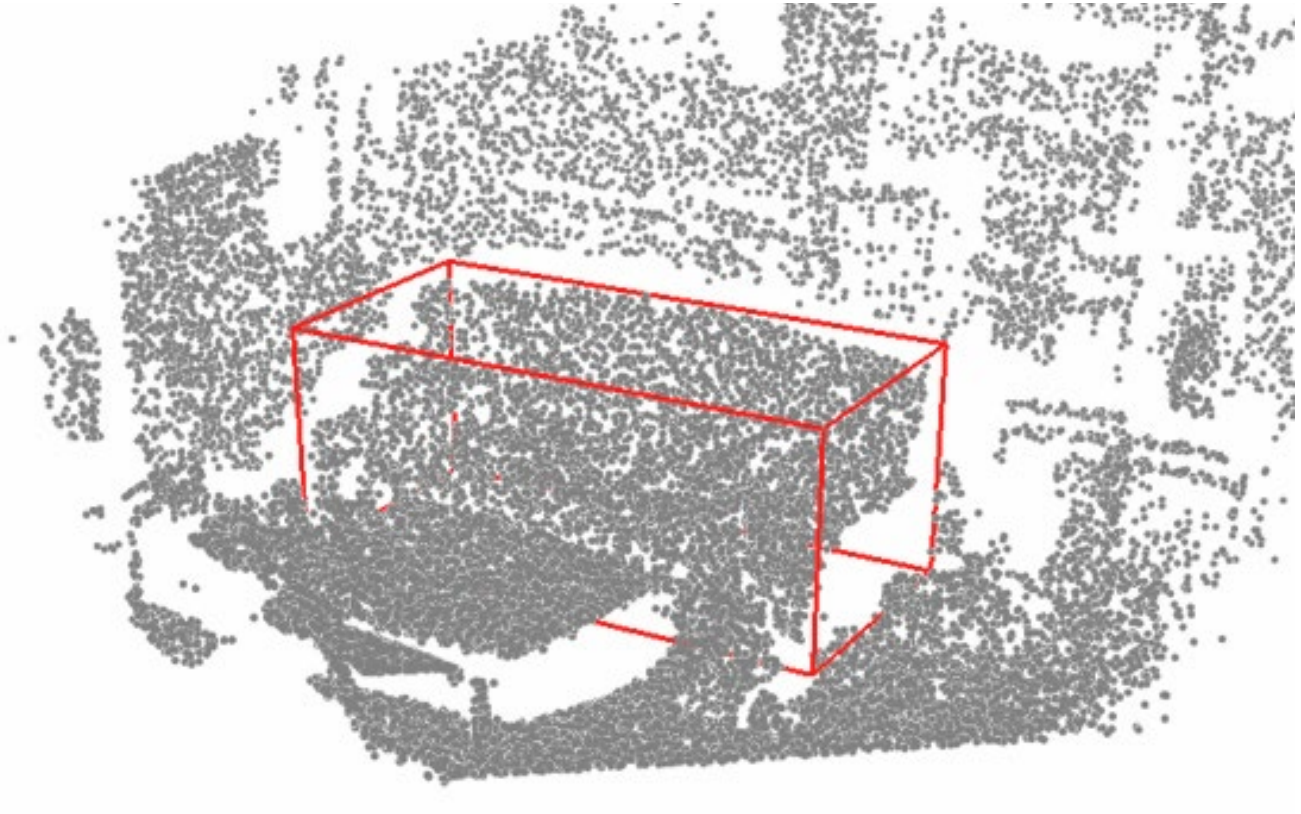
ScanNetV2

| | Input | mAP@0.25 | mAP@0.5 |
|---|---|---|---|
| DSS [42, 12] | Geo + RGB | 15.2 | 6.8 |
| MRCNN 2D-3D [11, 12] | Geo + RGB | 17.3 | 10.5 |
| F-PointNet [34, 12] | Geo + RGB | 19.8 | 10.8 |
| GSPN [54] | Geo + RGB | 30.6 | 17.7 |
| 3D-SIS [12] | Geo + 1 view | 35.1 | 18.7 |
| 3D-SIS [12] | Geo + 3 views | 36.6 | 19.0 |
| 3D-SIS [12] | Geo + 5 views | 40.2 | 22.5 |
| 3D-SIS [12] | Geo only | 25.4 | 14.6 |
| VoteNet (ours) | Geo only | **58.6** | **33.5** |

# Images and Point Clouds are Complementary



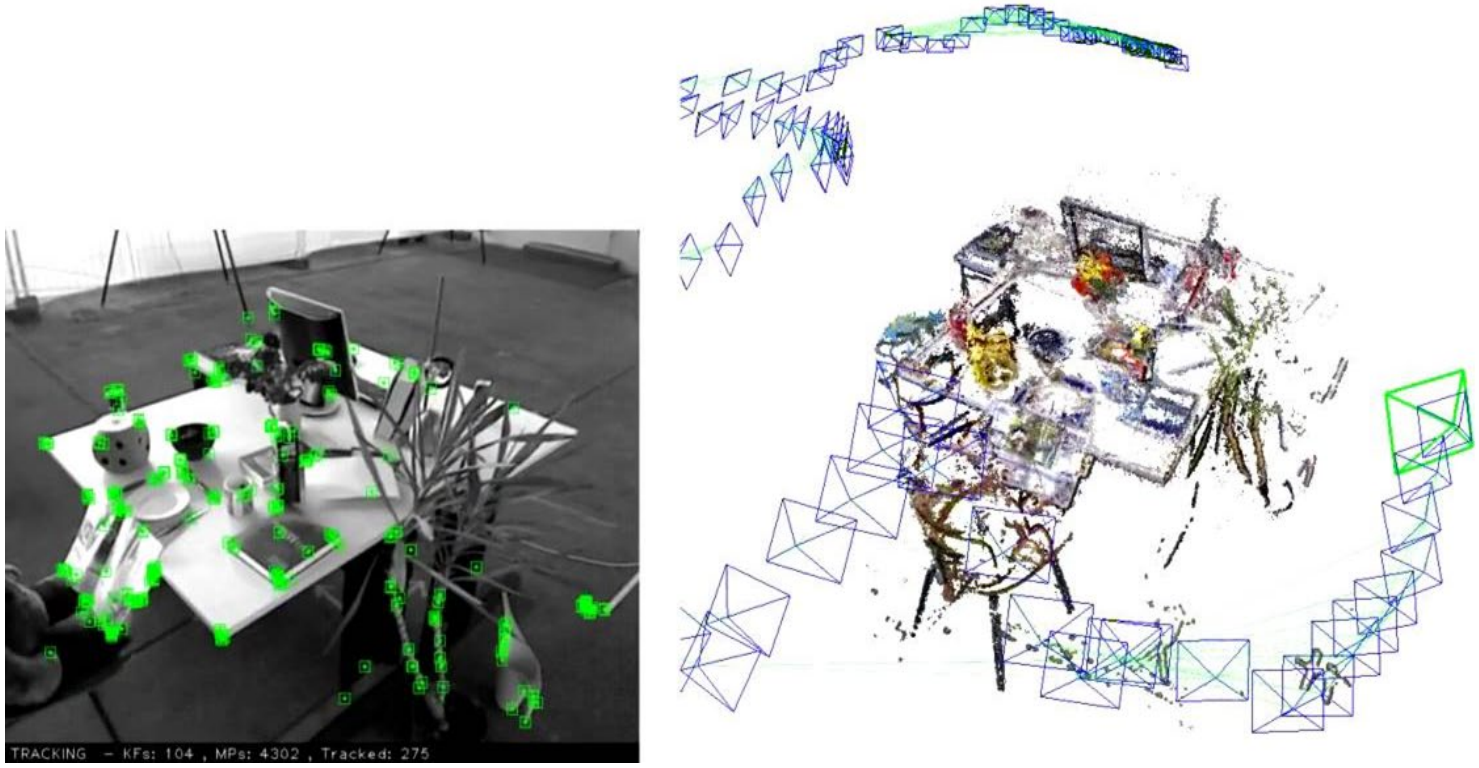
- High resolution
- See "blind regions"



- Absolute depth and scale

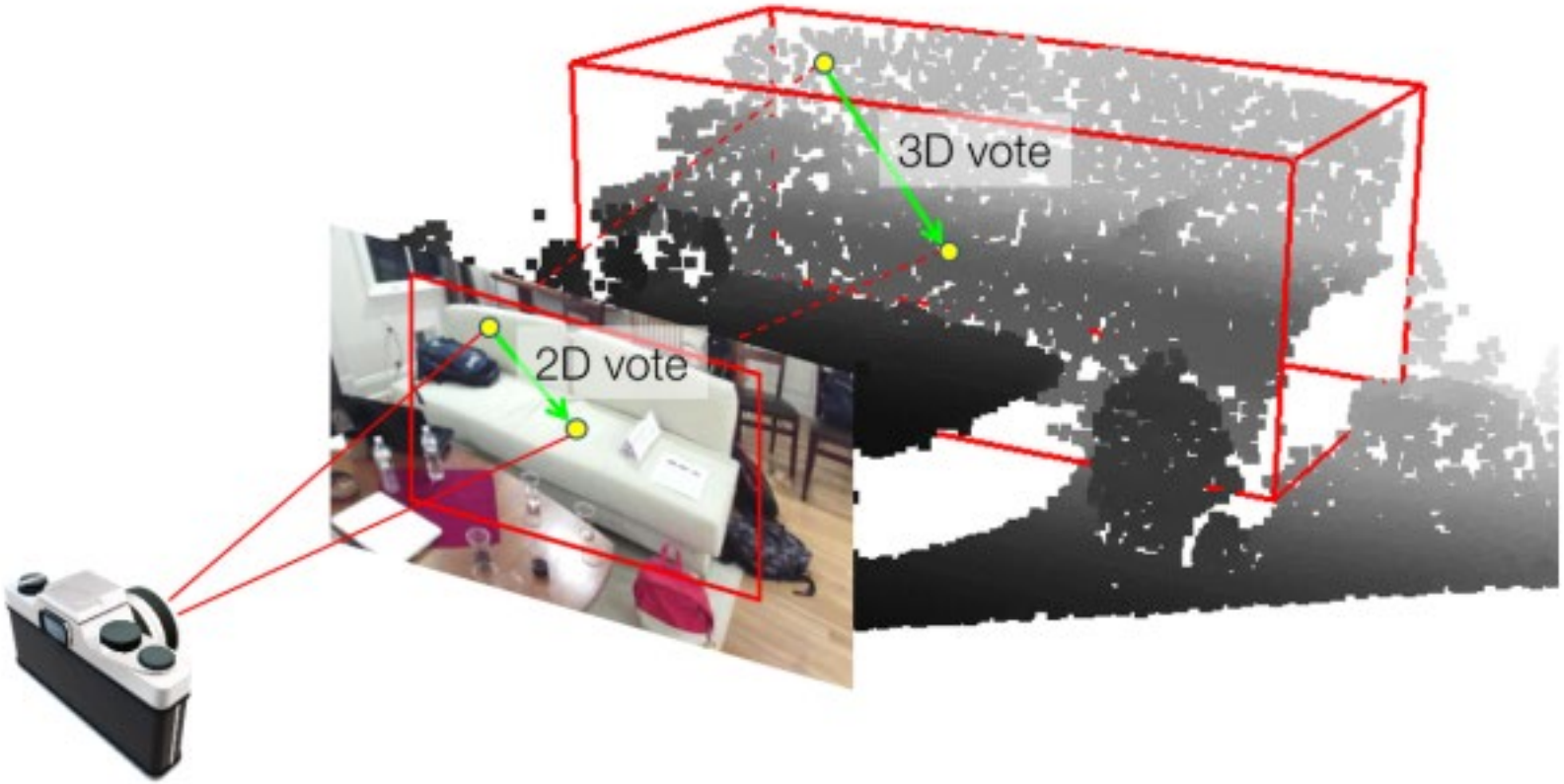
# 3D Detection with Sparse Points

**Application:** 3D detection from monocular video, using sparse SLAM keypoints.



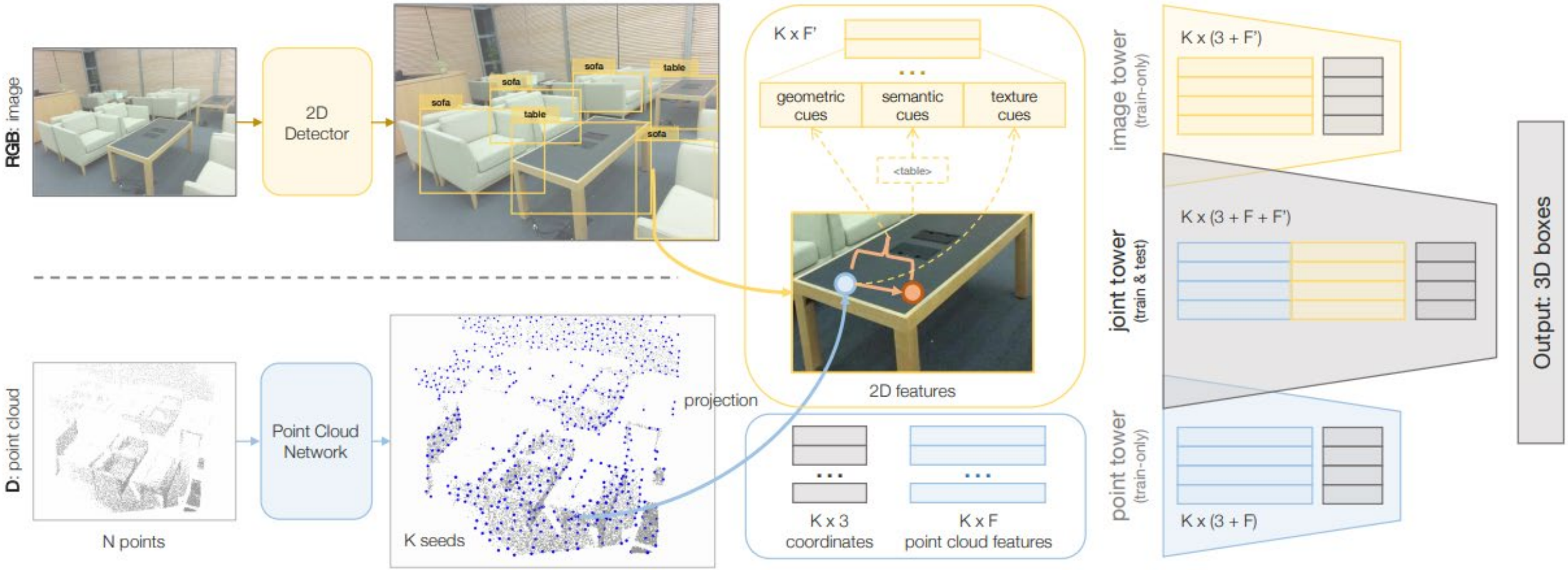

*Picture: ORB-SLAM results*

# Basic idea: *ImVoteNet*



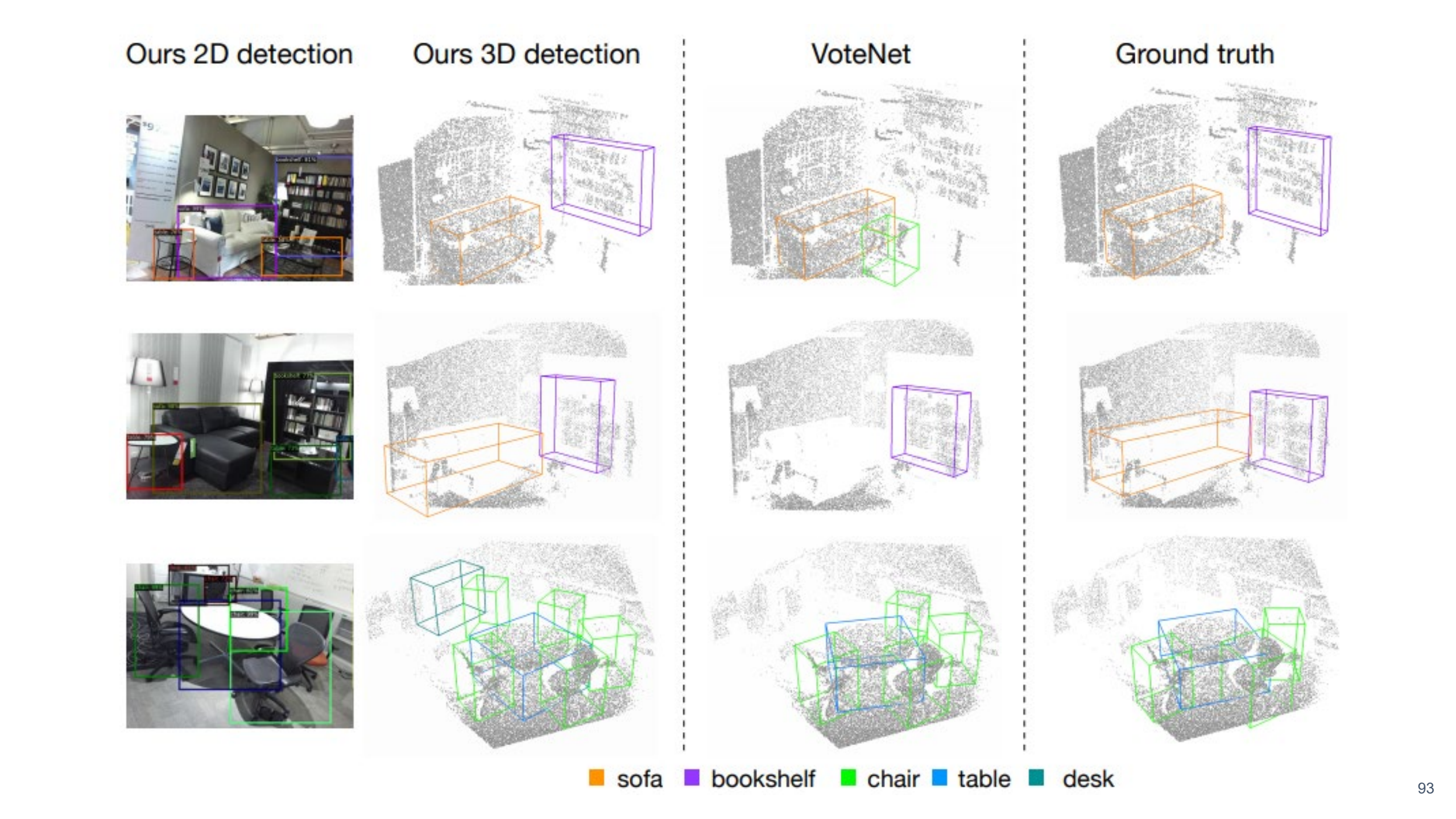

C. Qi, X. Chen, O. Litany, L.J. Guibas CVPR 2020

# *ImVoteNet* Architecture

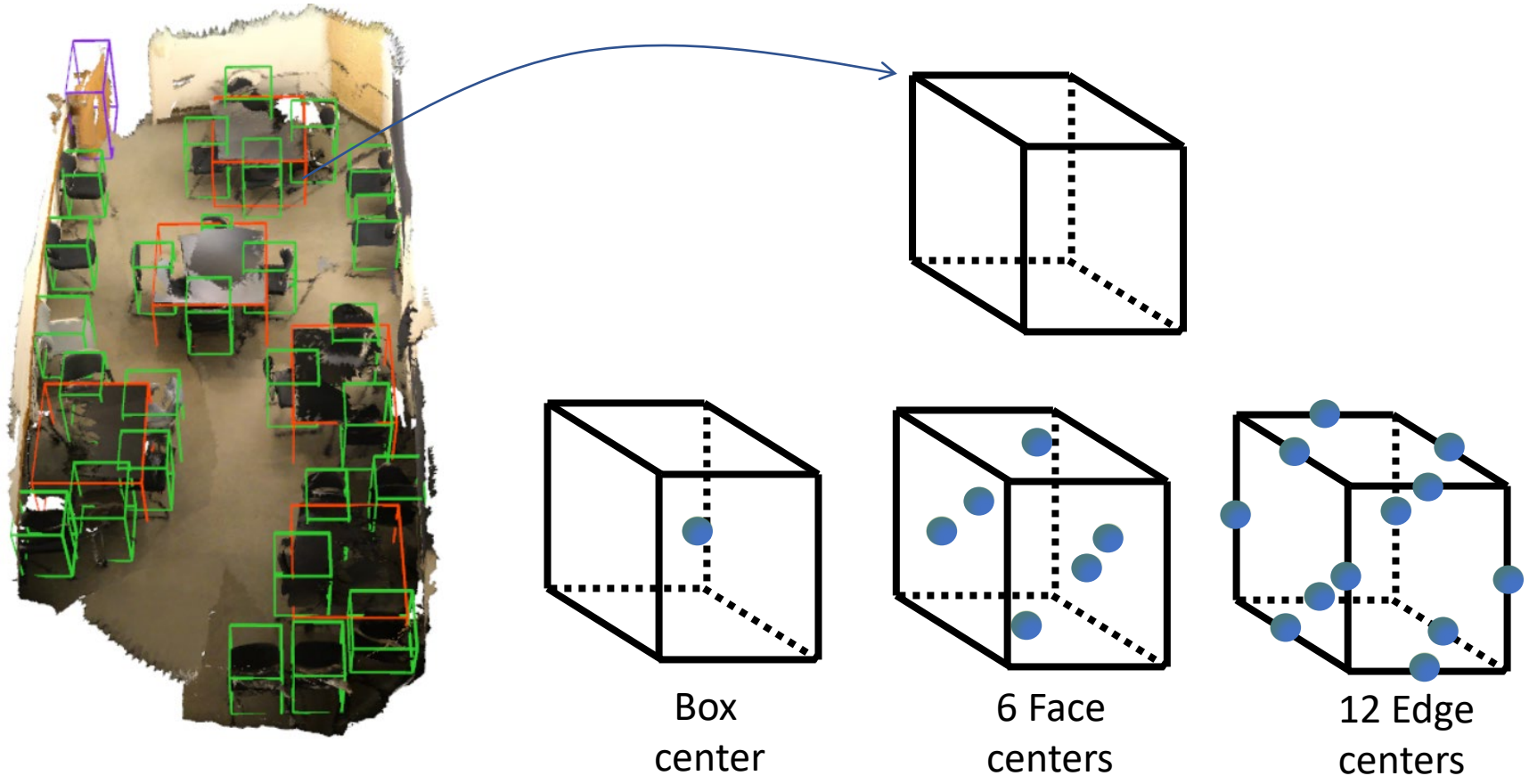


# Results on SUN RGB-D

| methods | RGB | bathtub | bed | bookshelf | chair | desk | dresser | nightstand | sofa | table | toilet | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DSS [39] | ✓ | 44.2 | 78.8 | 11.9 | 61.2 | 20.5 | 6.4 | 15.4 | 53.5 | 50.3 | 78.9 | 42.1 |
| COG [34] | ✓ | 58.3 | 63.7 | 31.8 | 62.2 | **45.2** | 15.5 | 27.4 | 51.0 | **51.3** | 70.1 | 47.6 |
| 2D-driven [15] | ✓ | 43.5 | 64.5 | 31.4 | 48.3 | 27.9 | 25.9 | 41.9 | 50.4 | 37.0 | 80.4 | 45.1 |
| PointFusion [43] | ✓ | 37.3 | 68.6 | 37.7 | 55.1 | 17.2 | 23.9 | 32.3 | 53.8 | 31.0 | 83.8 | 45.4 |
| F-PointNet [29] | ✓ | 43.3 | 81.1 | 33.3 | 64.2 | 24.7 | 32.0 | 58.1 | 61.1 | 51.1 | **90.9** | 54.0 |
| VOTENET [28] | ✗ | 74.4 | 83.0 | 28.8 | 75.3 | 22.0 | 29.8 | 62.2 | 64.0 | 47.3 | 90.1 | 57.7 |
| +RGB | ✓ | 70.0 | 82.8 | 27.6 | 73.1 | 23.2 | 27.2 | 60.7 | 63.7 | 48.0 | 86.9 | 56.3 |
| +region feature | ✓ | 71.7 | 86.1 | 34.0 | 74.7 | 26.0 | 34.2 | 64.3 | 66.5 | 49.7 | 88.4 | 59.6 |
| IMVOTENET | ✓ | **75.9** | **87.6** | **41.3** | **76.7** | 28.7 | **41.4** | **69.9** | **70.7** | 51.1 | 90.5 | **63.4** |

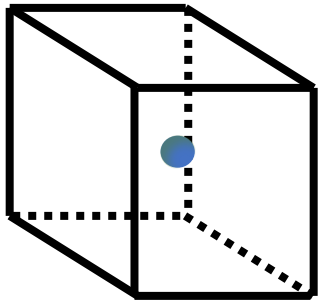

Ours 2D detection
Ours 3D detection
VoteNet
Ground truth
sofa
bookshelf
chair
table
desk

# Multiple 3D Geometry Representations

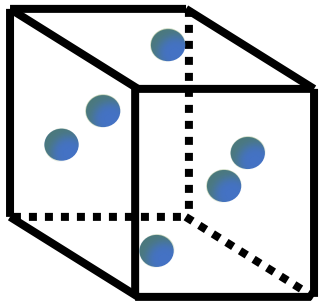



[Z. Zhang, B. Sun, H. Yang, Q. Huang.
**H3DNet: 3D Object Detection Using Hybrid Geometric Primitives**.
ECCV 2020]

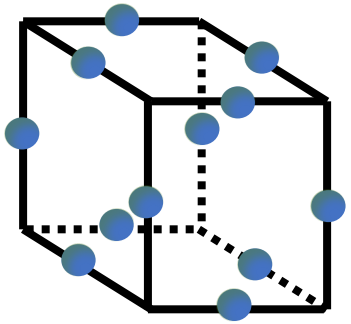
# Representations Best for Different Object Instances



Box
center



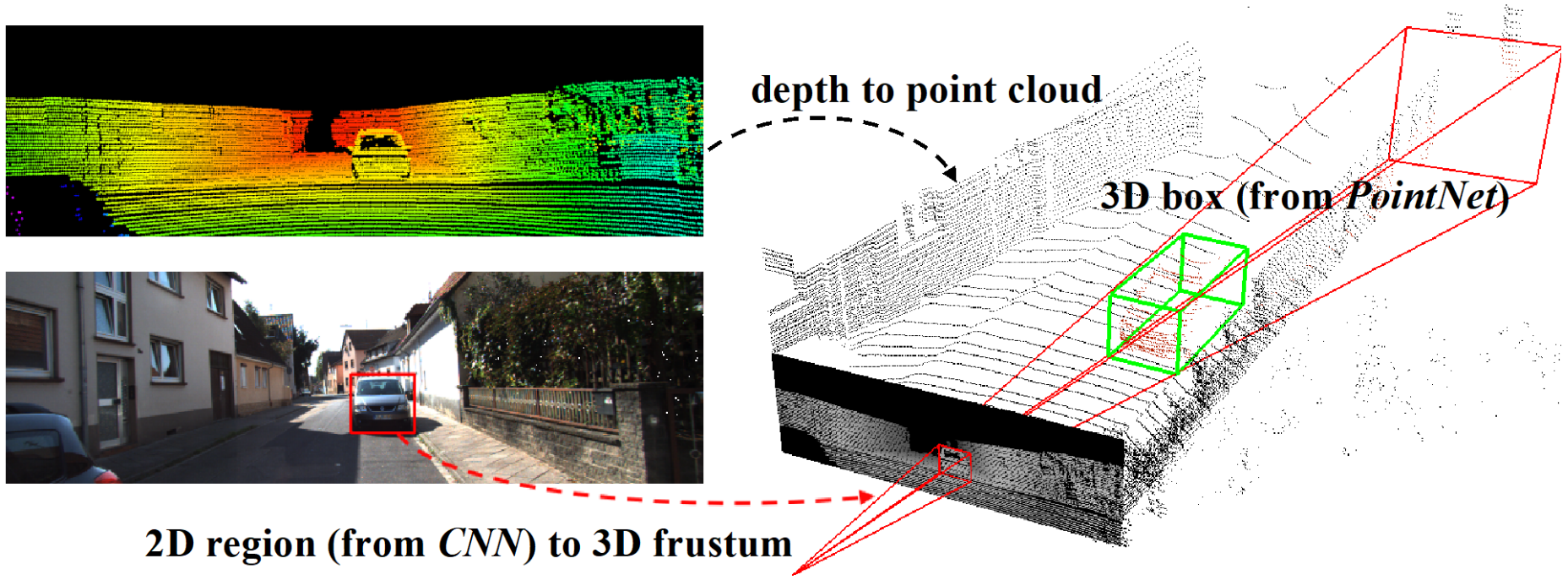
6 Face
centers



12 Edge
centers

# Object Detection in Point Clouds, Outdoors

# Frustum PointNets for 3D Object Detection

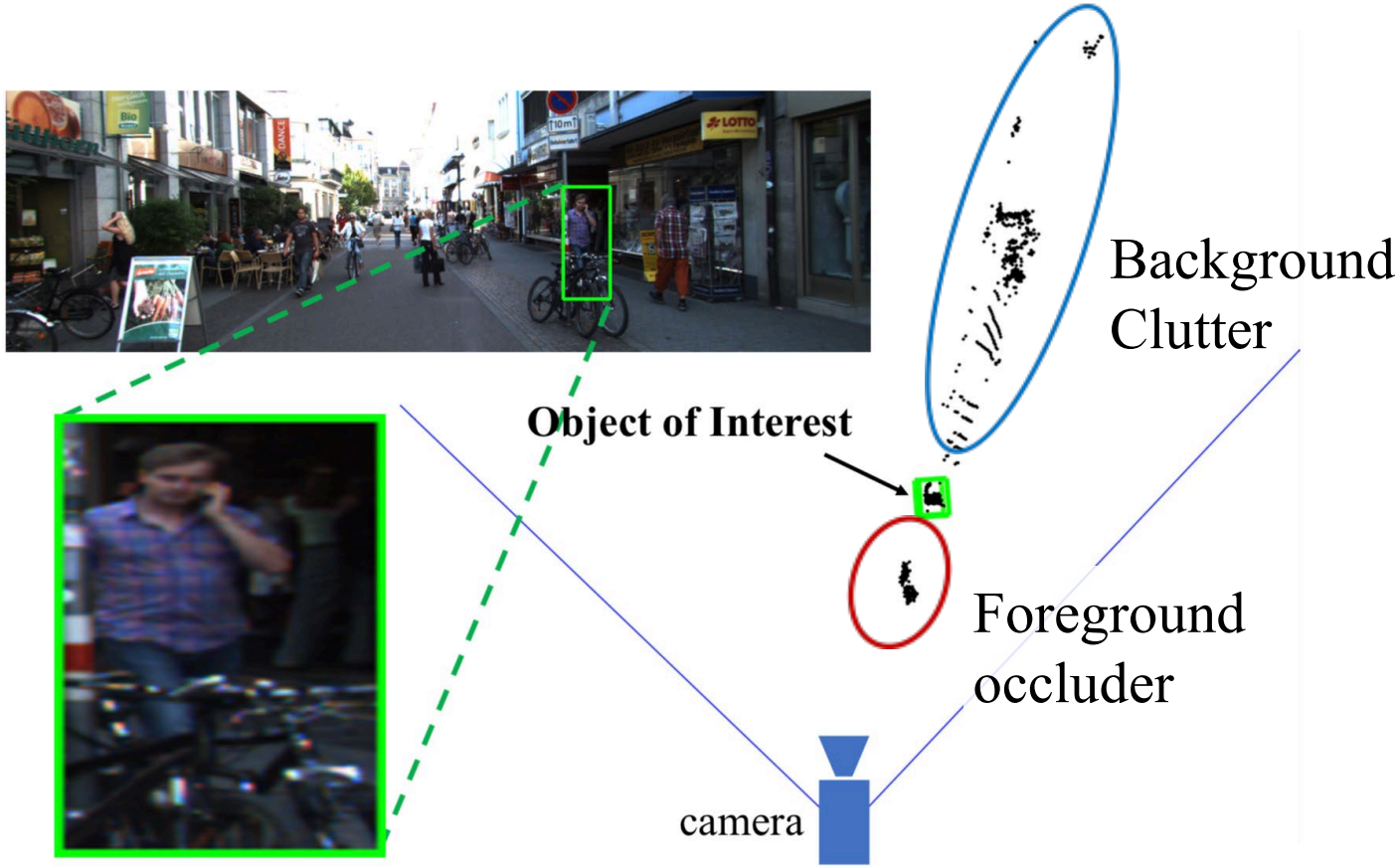



***+ Leveraging mature 2D detectors for region proposal. greatly reducing 3D search space.***

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

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

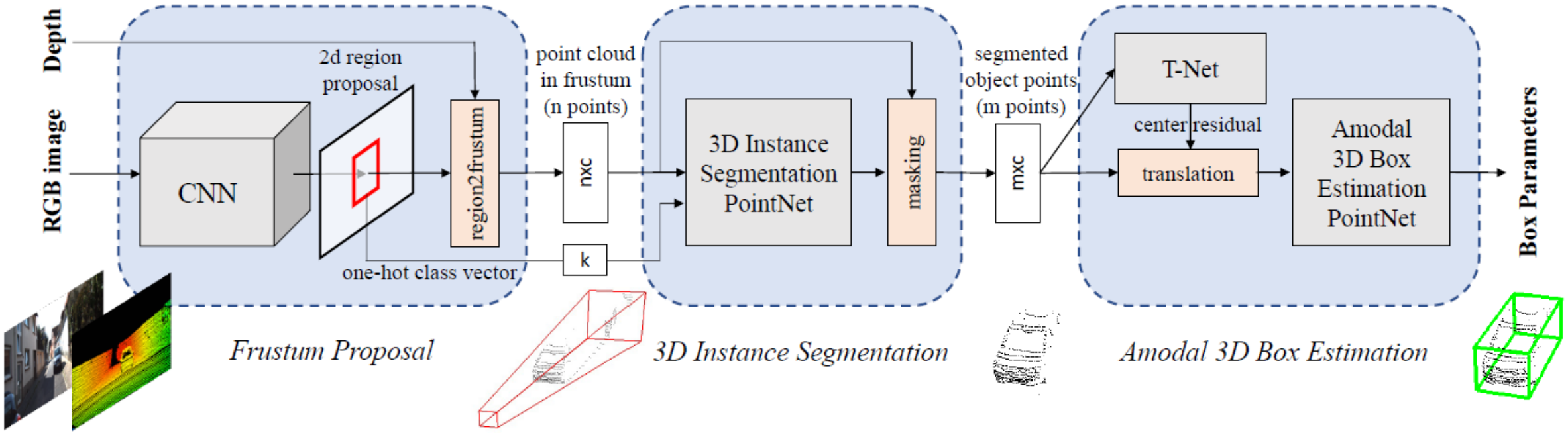
# Frustum-based 3D Object Detection: Challenges




- Occlusion and clutter is common in frustum point clouds
- Large range of point depths

# Frustum PointNets

Use **PointNets** for **data-driven** object detection in frustums.




- pose
- size
- center

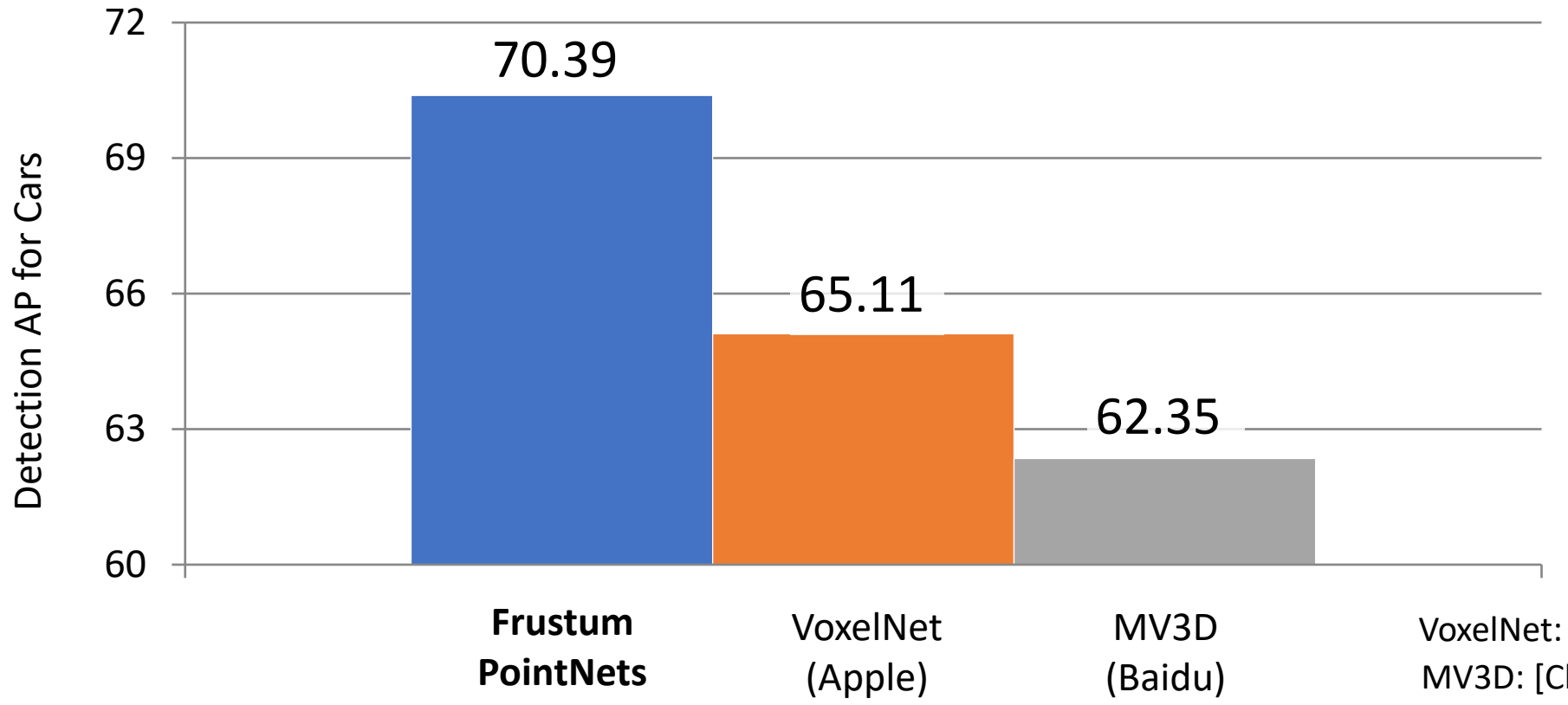
# Frustum PointNets: Key to Success

***Respect and exploit 3D***

- **Use each modality (image, points) for what it's best at** — using 3D representation and 3D deep learning for the 3D problem.
- **Canonicalize the problem** — exploiting geometric transformations in point clouds.

# KITTI Results: Quantitative

*Leading performance on KITTI benchmark*

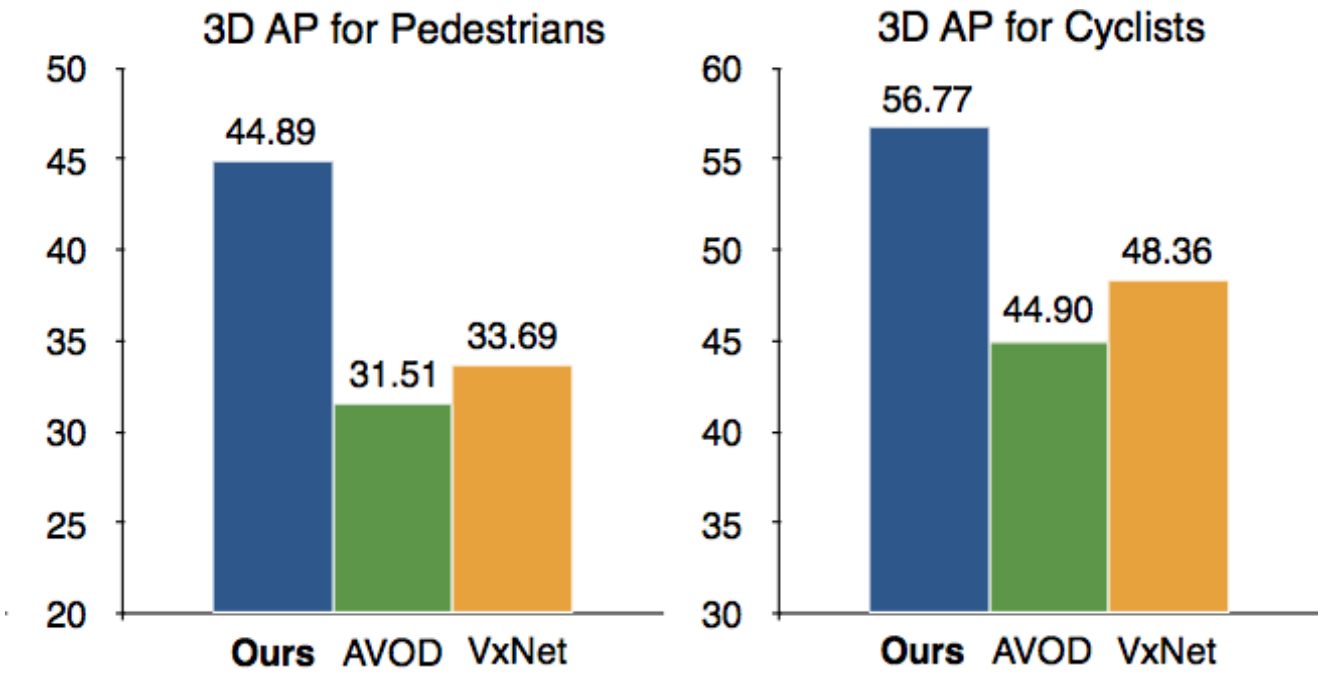



VoxelNet: [Zhou et al. 2018]
MV3D: [Chen et al. 2017]

# KITTI Results: Quantitative

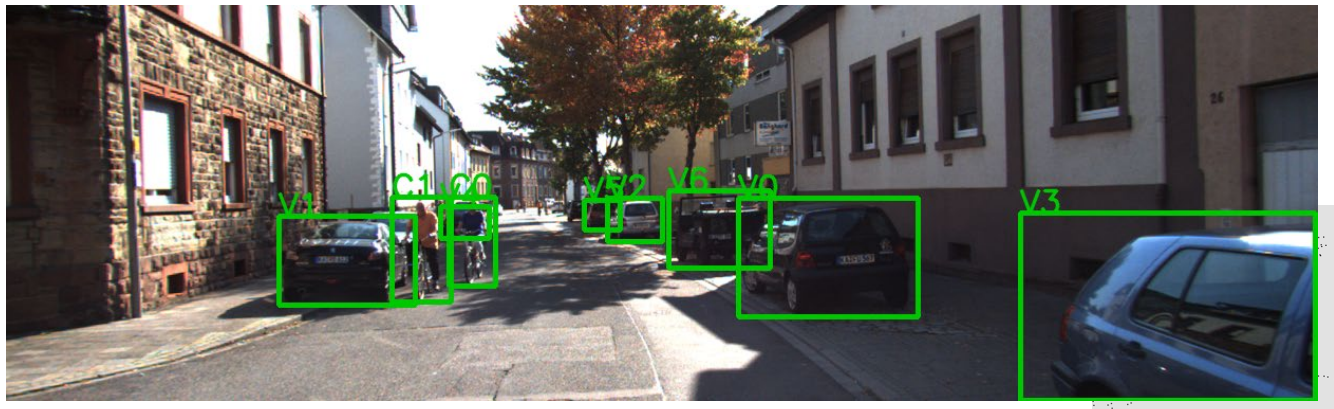
*Leading performance on KITTI benchmark*

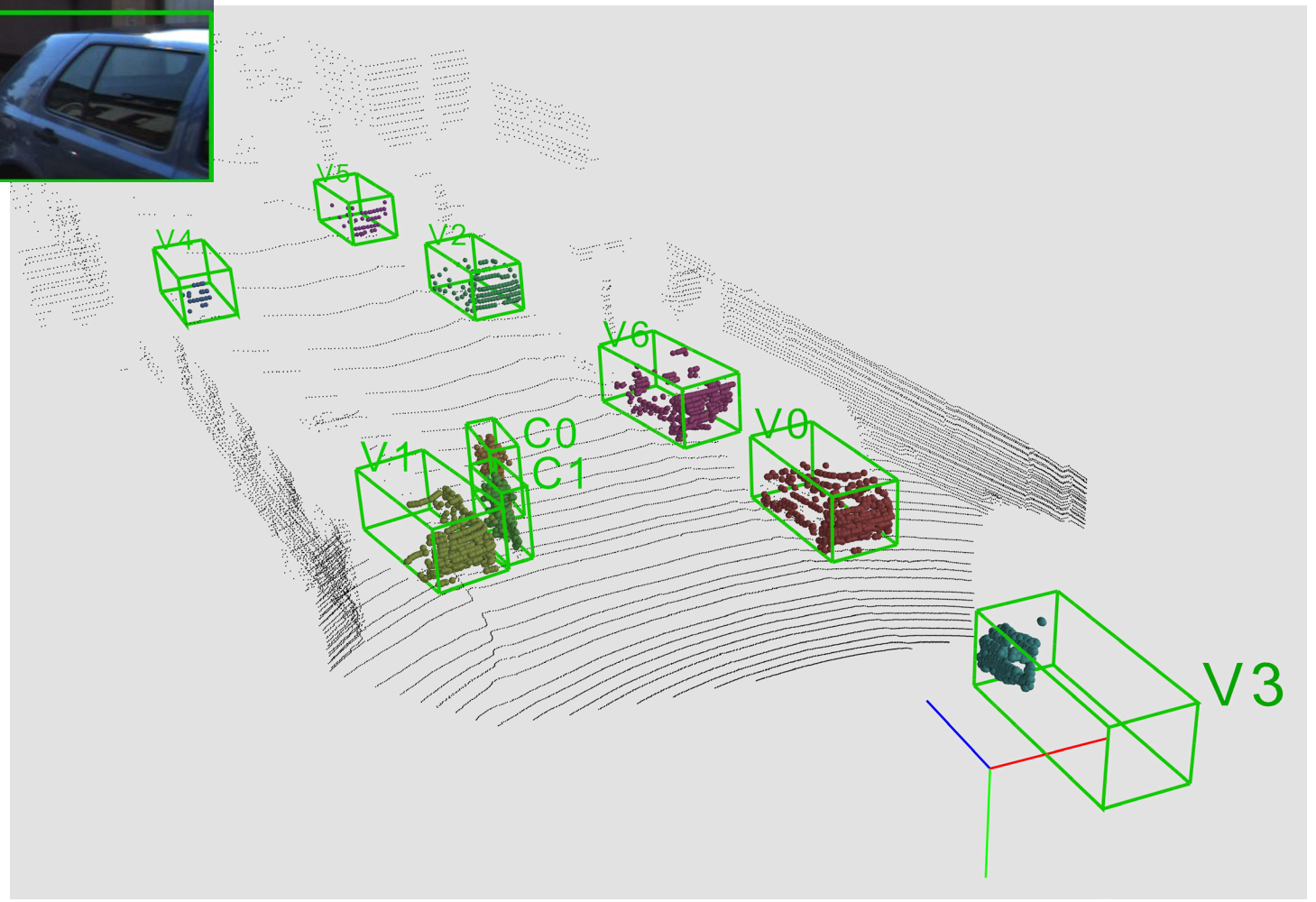
Especially leading at smaller objects (pedestrians and cyclists)
– hard to localize with 3D proposals only.




AVOD: [Ku et al. 2018]
VxNet: [Zhou et al. 2017]

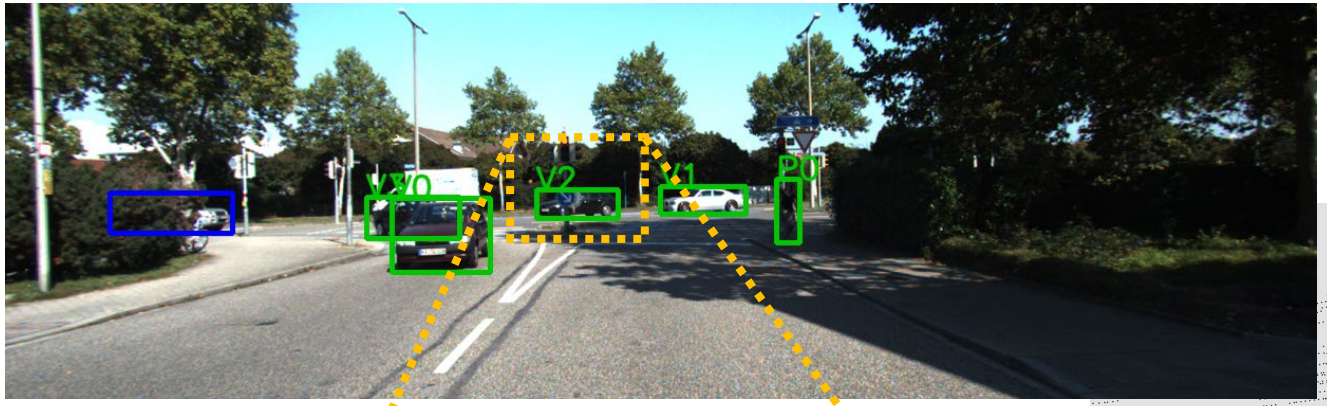
# KITTI Results: Qualitative

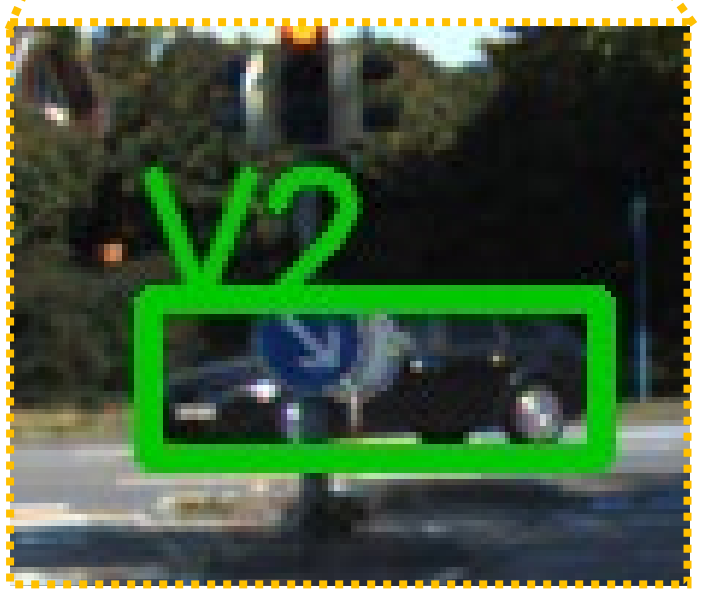


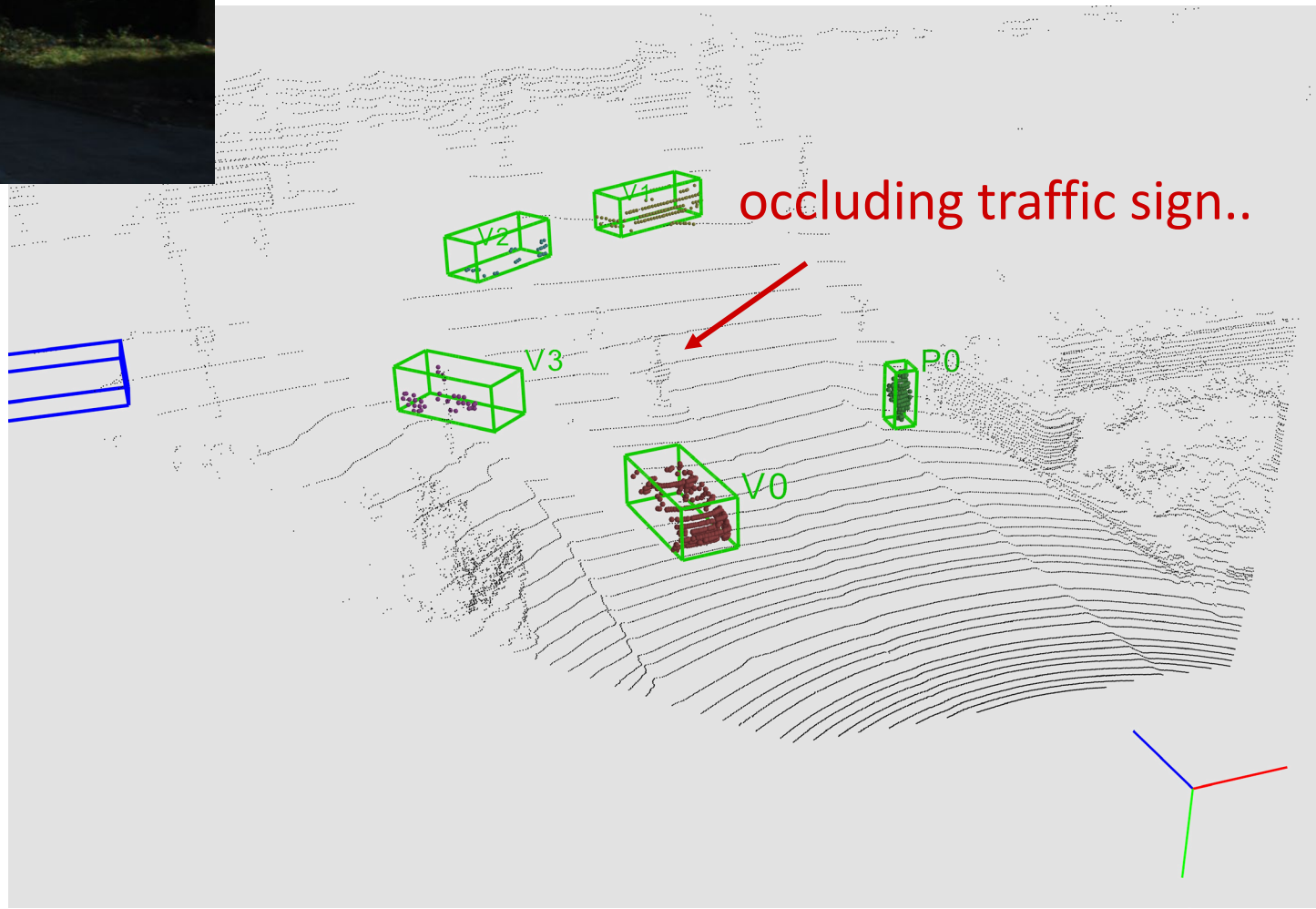

Remarkable box estimation accuracy even with a dozen of points or with very partial point clouds.

# KITTI Results: Qualitative

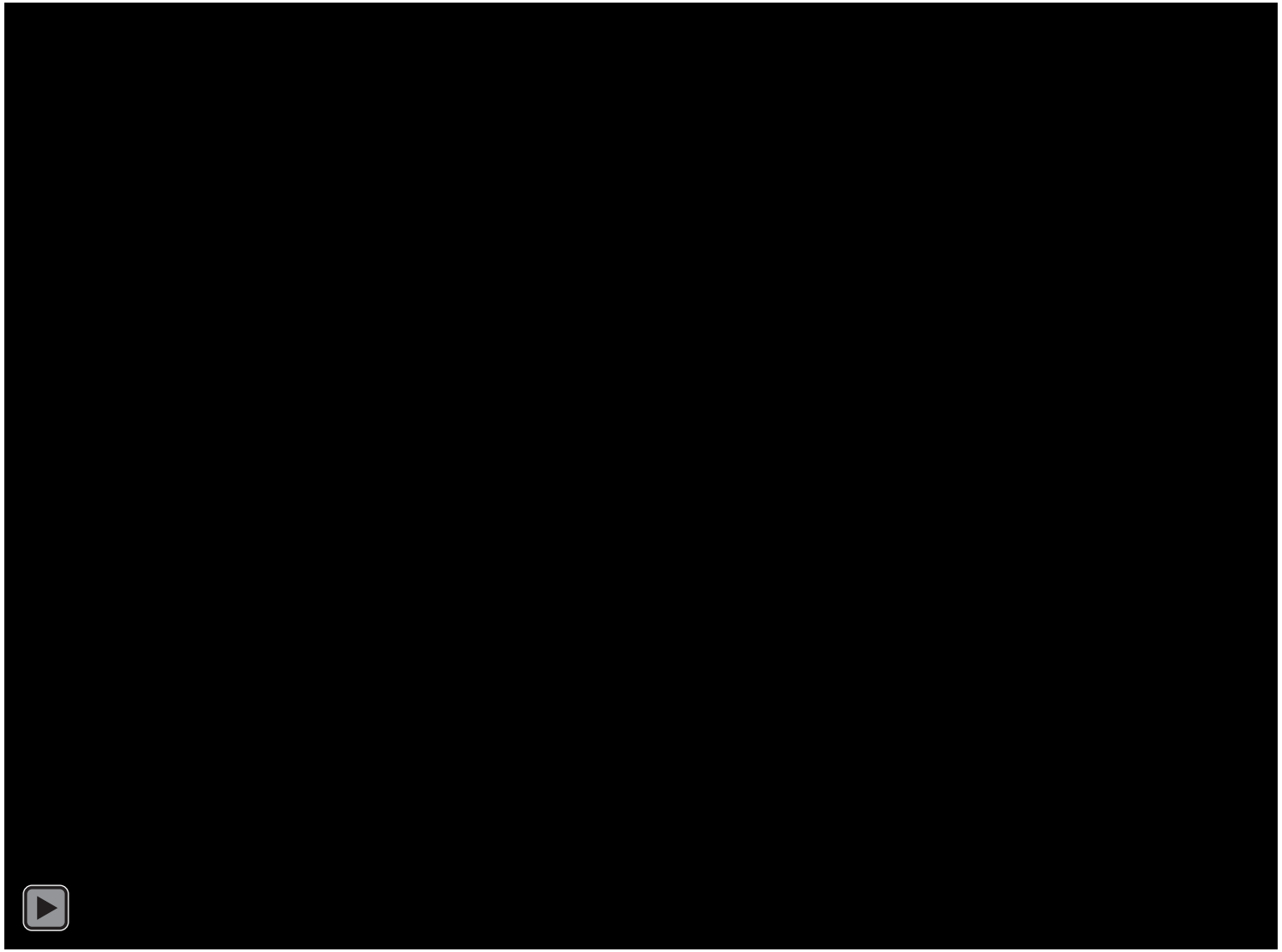



Correct segmentation in point clouds with heavy occlusion.



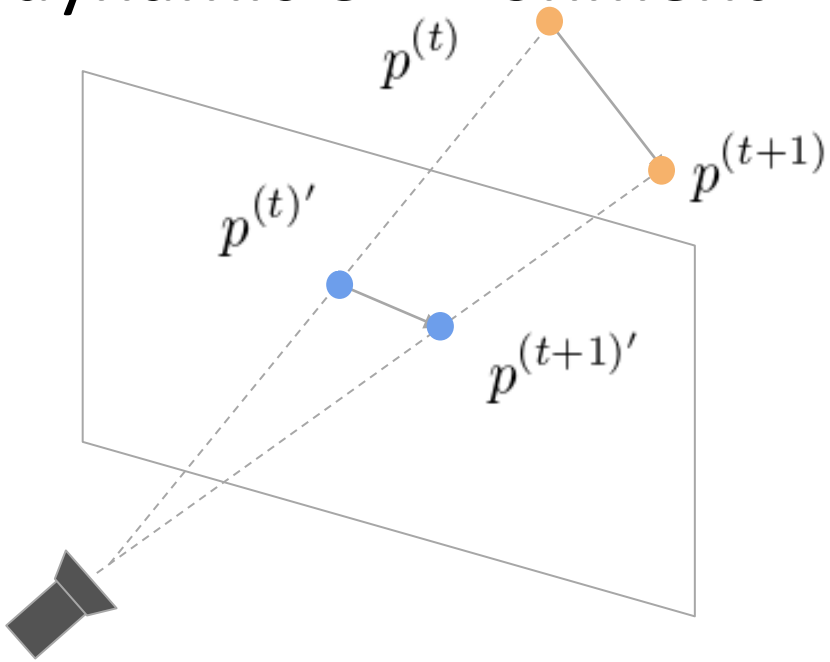
# KITTI Results: Example



# 3D Motion in Point Clouds

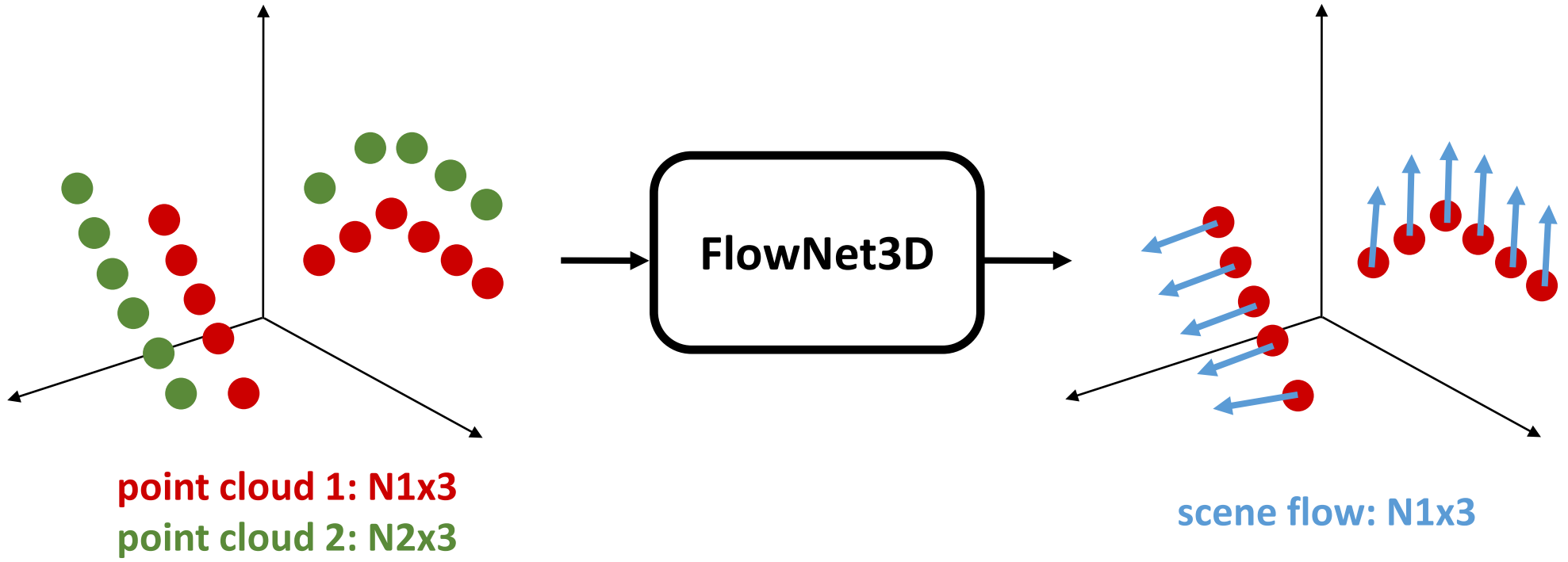
# Scene Flow [Vedula et al. 1999]

- Scene flow: 3D motion field of points
- Optical flow is its projection to 2D image plane.
- Low-level understanding of a dynamic environment



# Our Approach: FlowNet3D

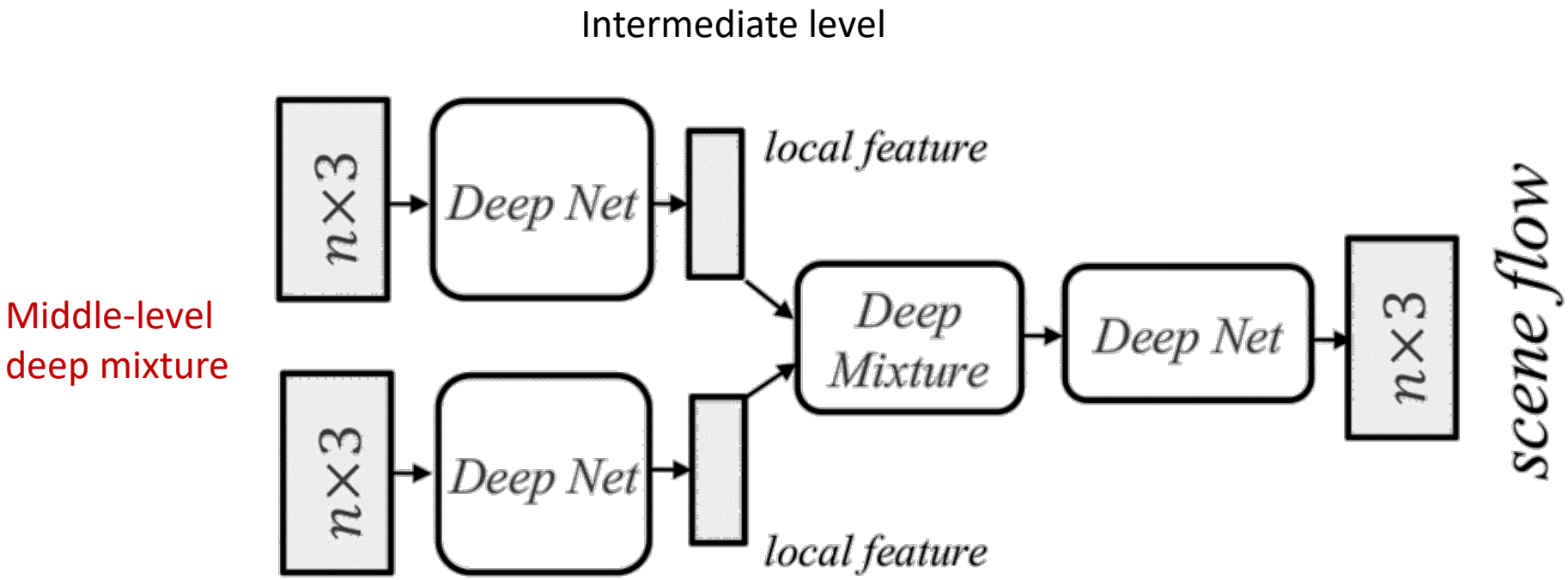
- Directly learning scene flow in 3D point clouds, with 3D deep learning architectures.



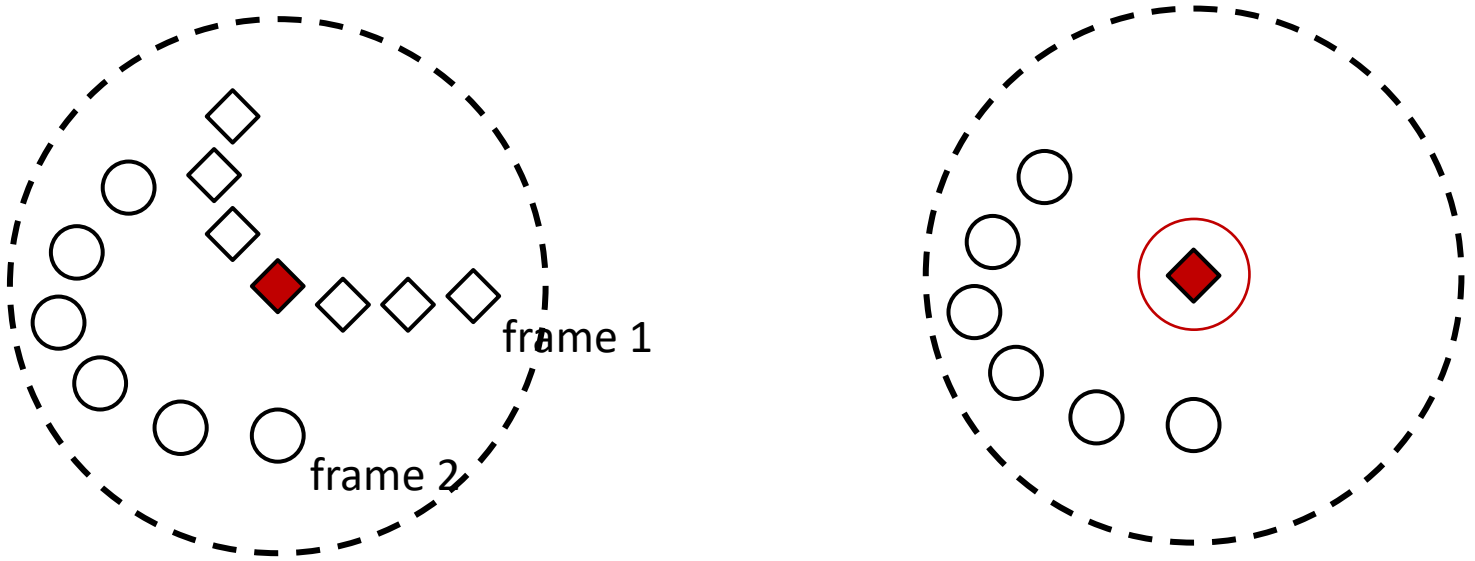

*Xingyu Liu, Charles R. Qi, Leonidas Guibas. Learning Scene Flow in 3D Point Clouds, (CVPR 2019).*

# Deep Net Architecture

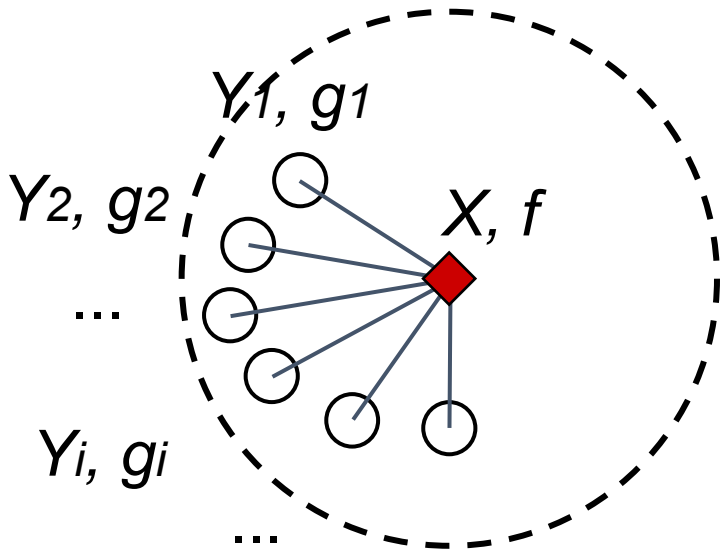
- How to learn point cloud features?
- Where in the network architecture to mix point features from consecutive frames?
- How to mix them?



# Middle-Level Mixing



# Point Attributes




dist($g_1$, f), $Y_1$-X

dist($g_2$, f), $Y_2$-X

⋮

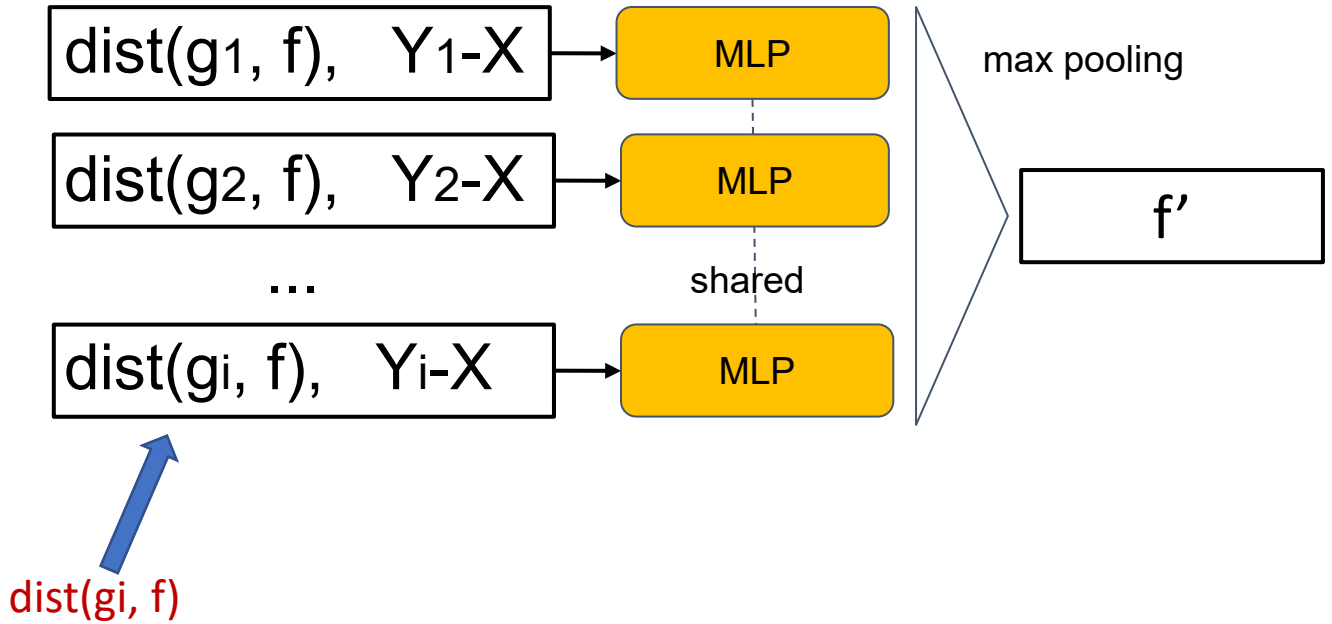
dist($g_i$, f), $Y_i$-X

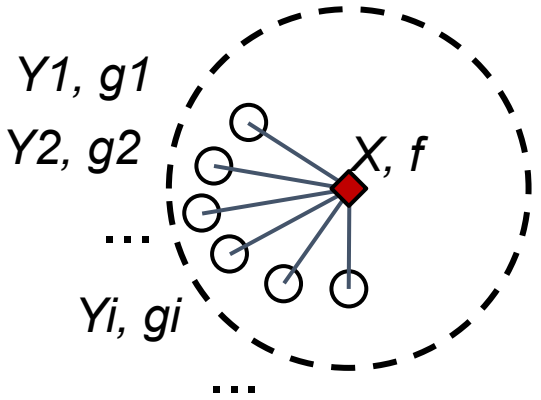
⋮

*Naive approach: concatenation*

| dist($g_1$, f), $Y_1$-X | dist($g_2$, f), $Y_2$-X | ... |
|---|---|---|

# A More Structured Approach



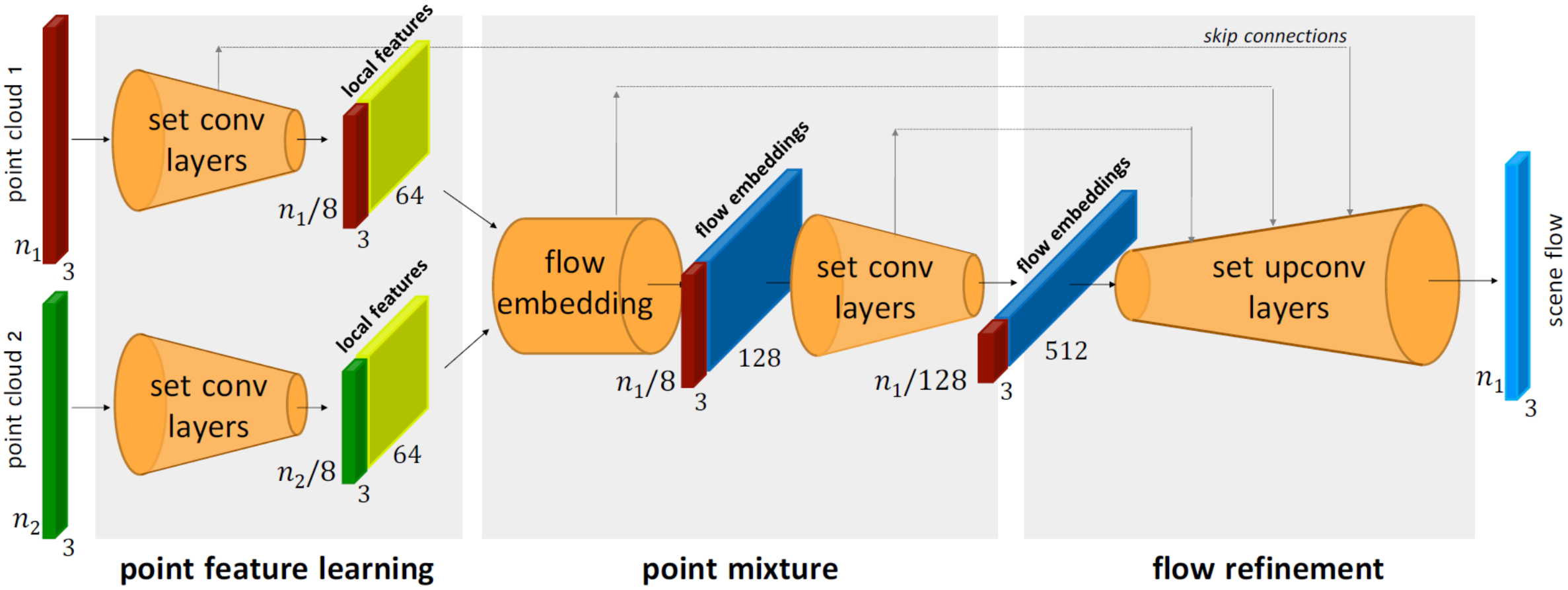




"Distance" functions:

- Euclidean distance (scalar)
- Cosine distance (scalar)
- Element-wise product (vector)

Let the network learn the distance function …

# FlowNet3D




set conv = set abstraction

Composed of many many mini-pointnet++ modules …

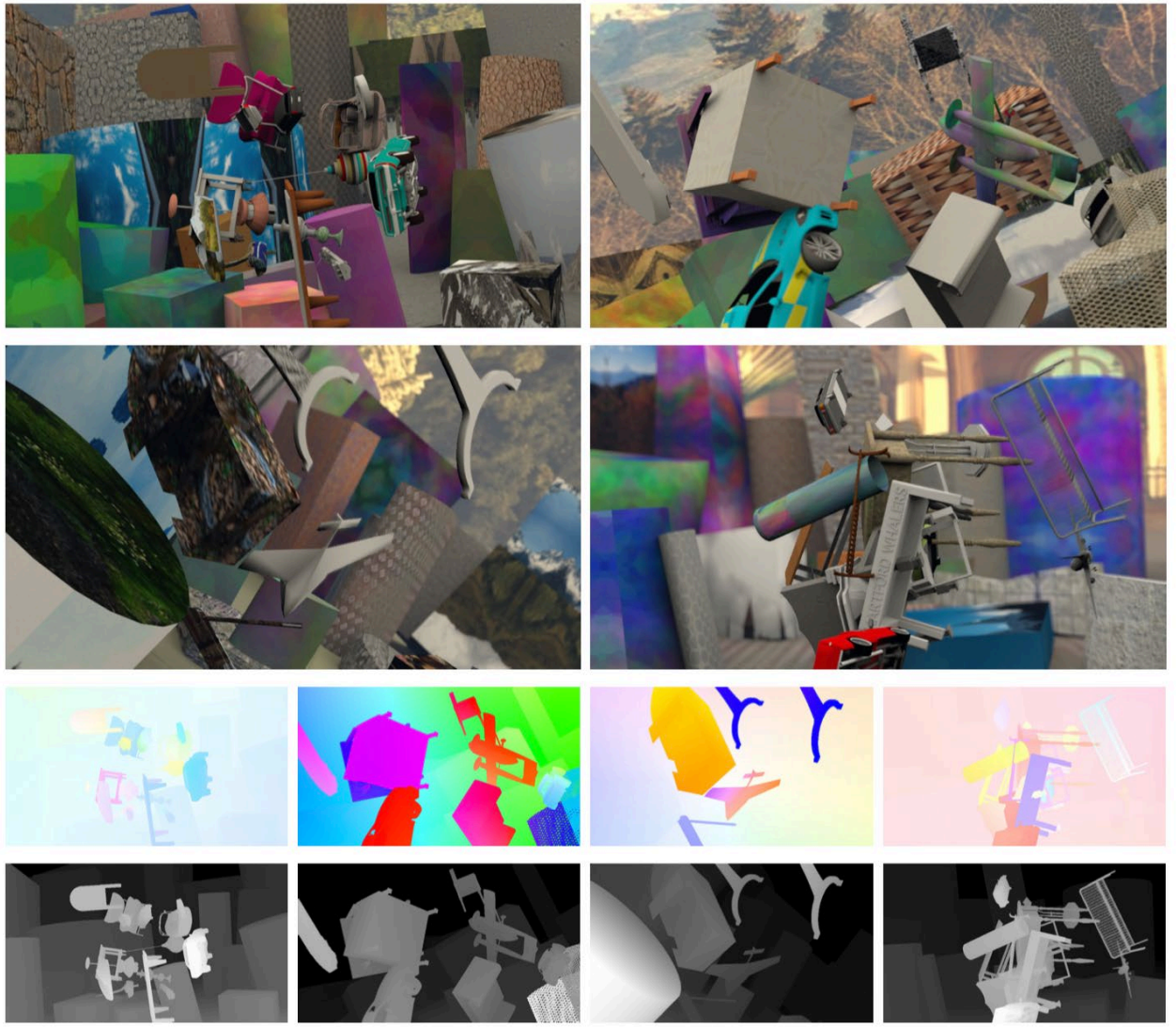
Pointnet++

# Training on Synthetic Data

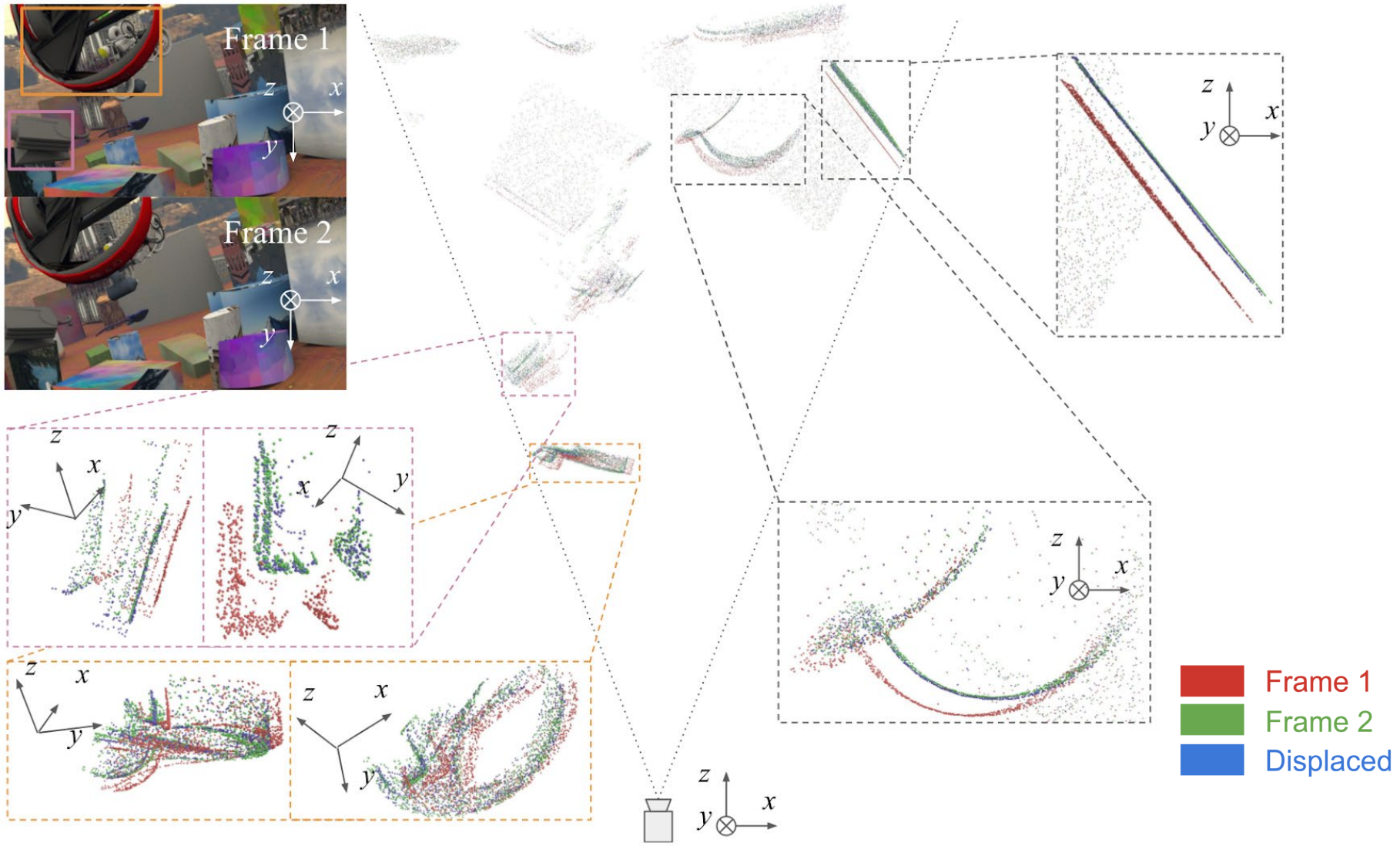
FlyingThings3D [Mayer et al. 2016] dataset from MPI

Random ShapeNet objects

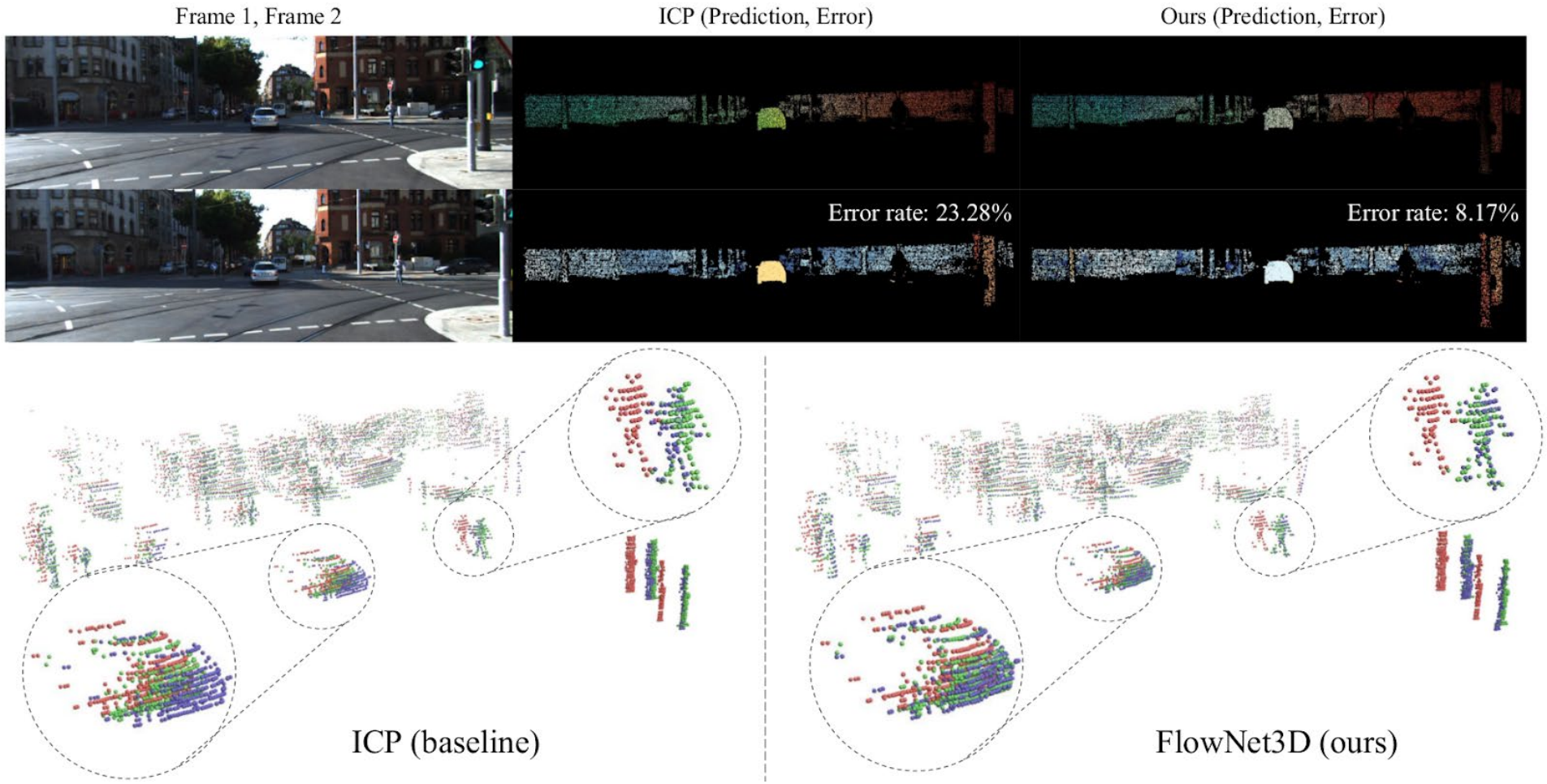
Very challenging dataset with strong occlusions and large motions.



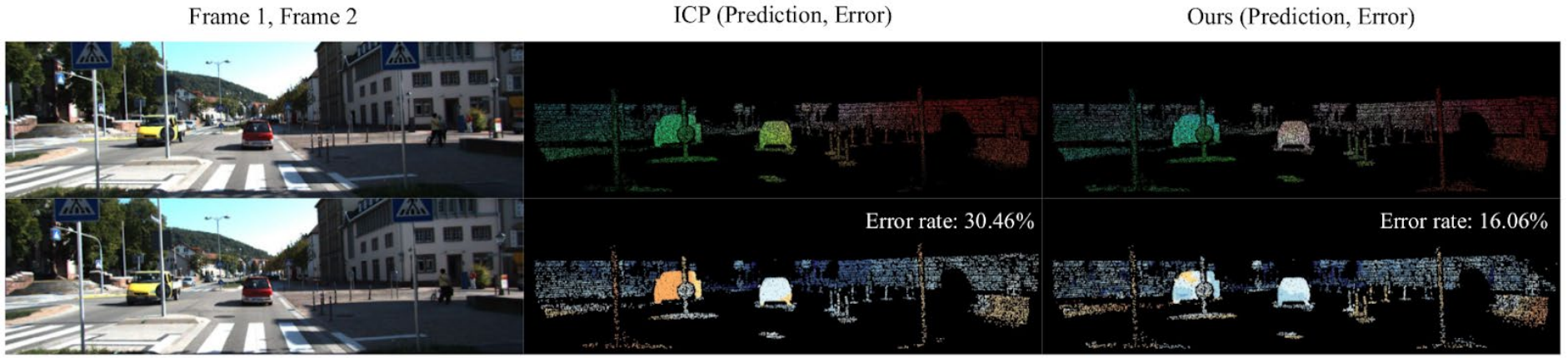
# FlyingThings3D Results

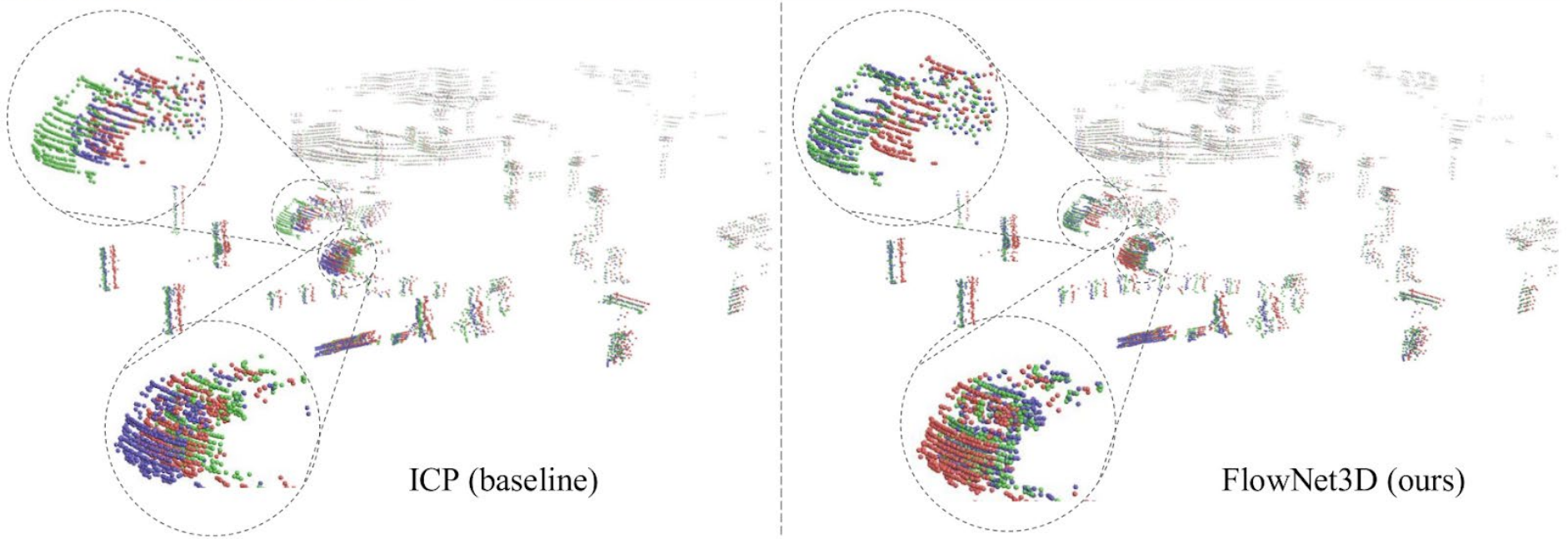


# KITTI Results

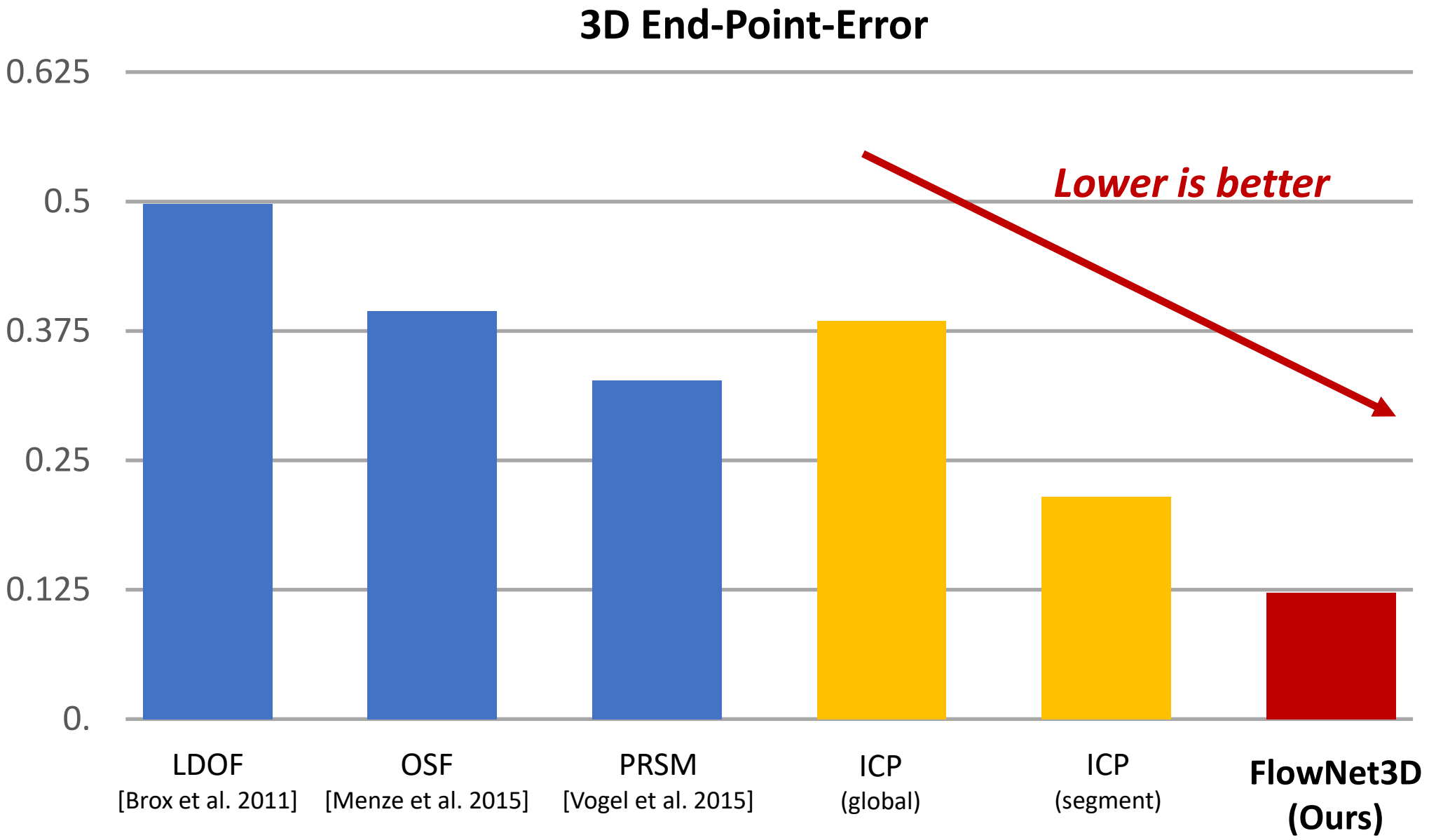


# KITTI Results



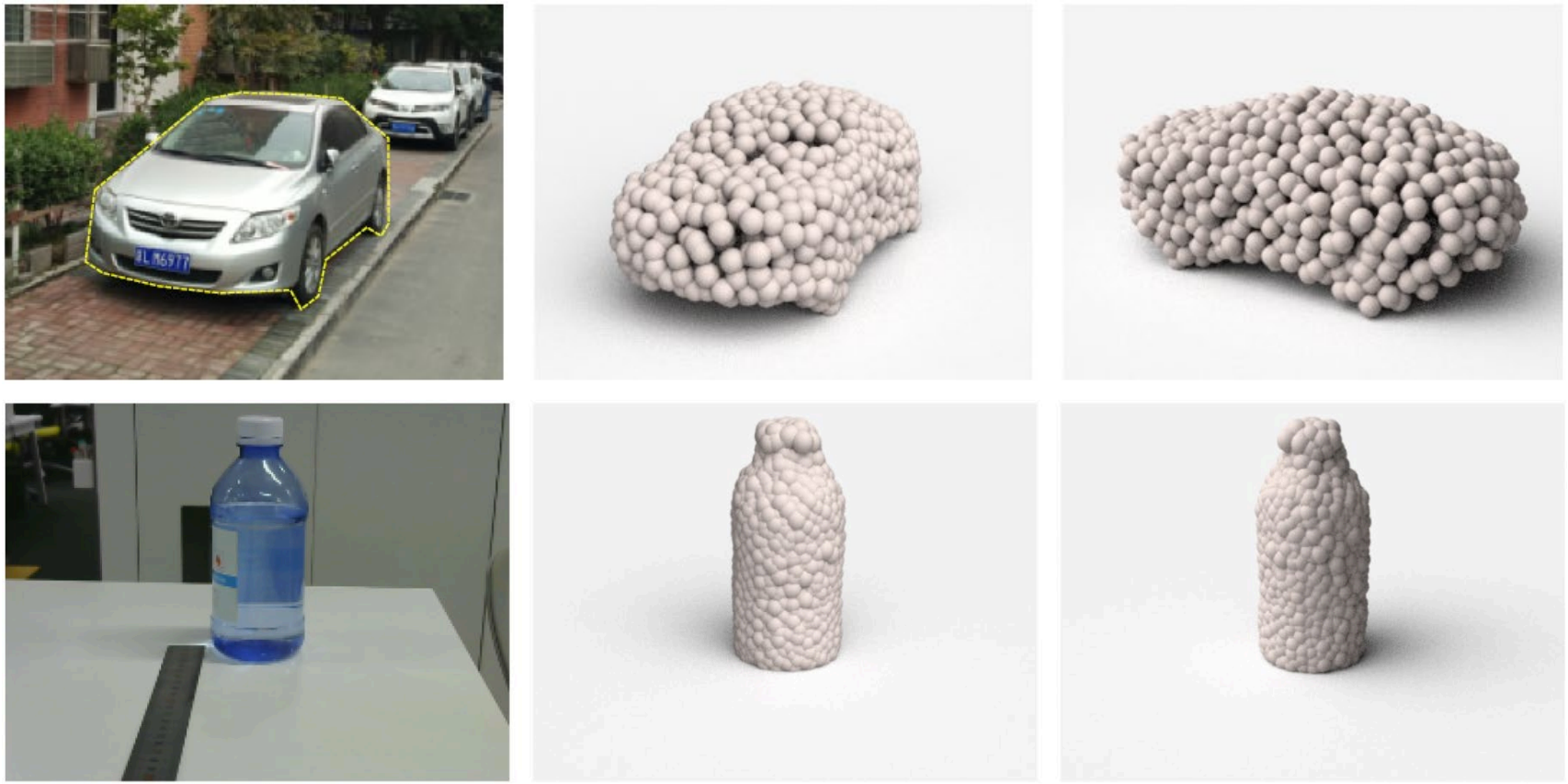



# Generalizing to KITTI: Quantitative



# Point-Set Generation

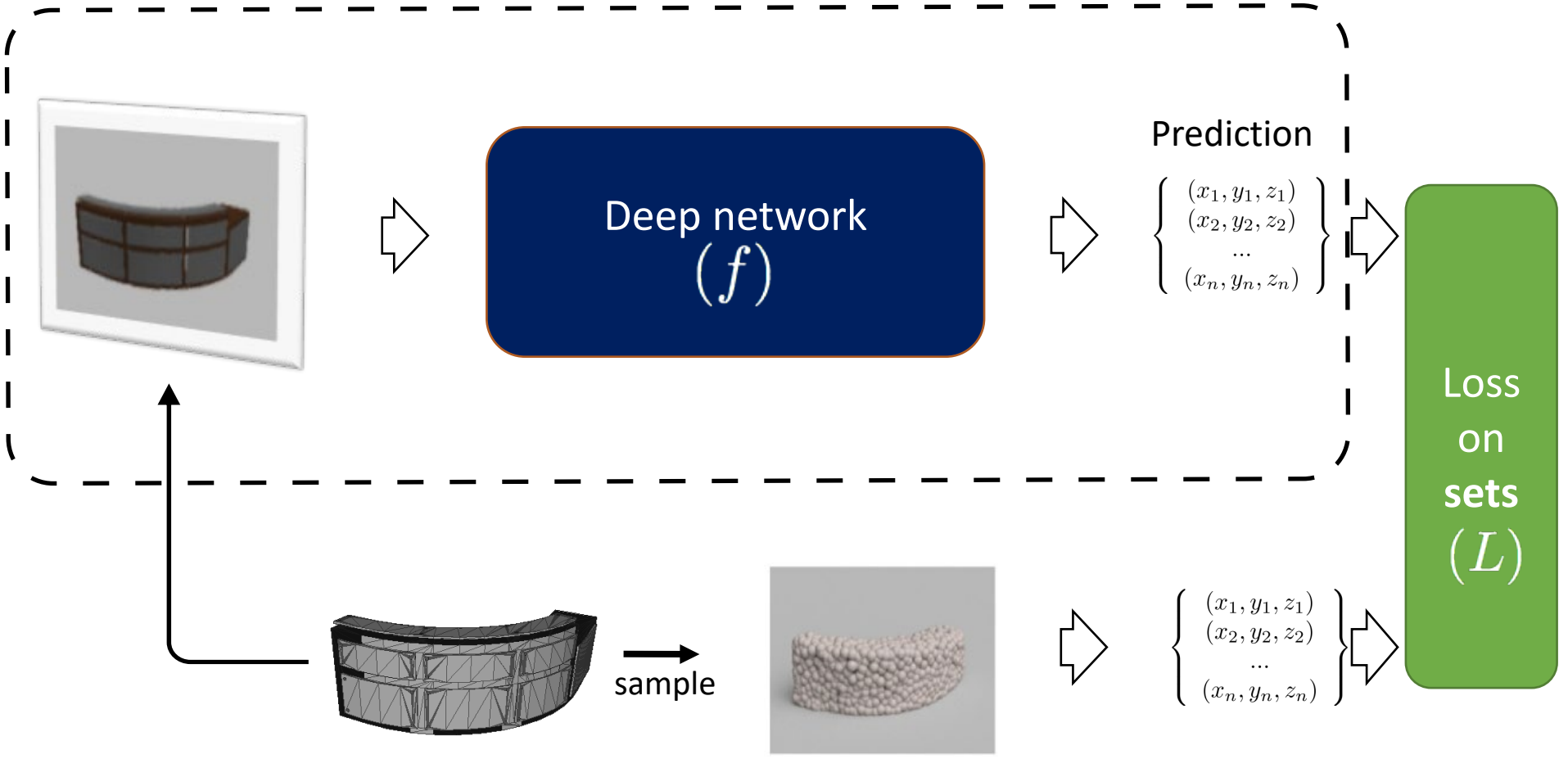
# Point Cloud Synthesis from a Single Image



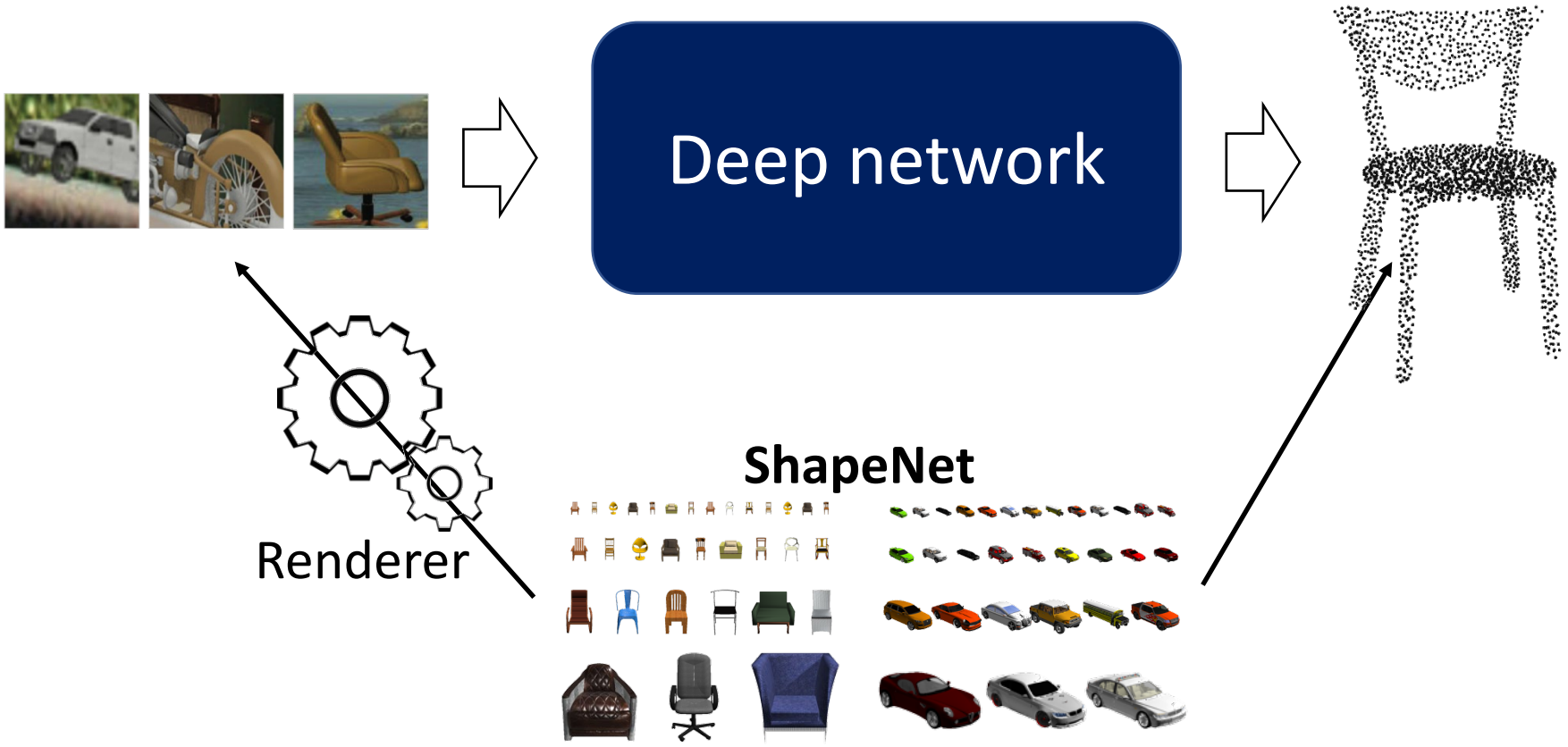
Input Reconstructed 3D point cloud

[H. Su, H. Fan, LG, 2017]

# End-to-End Learning



# Synthesize for Learning



# Point Cloud Distance Metrics

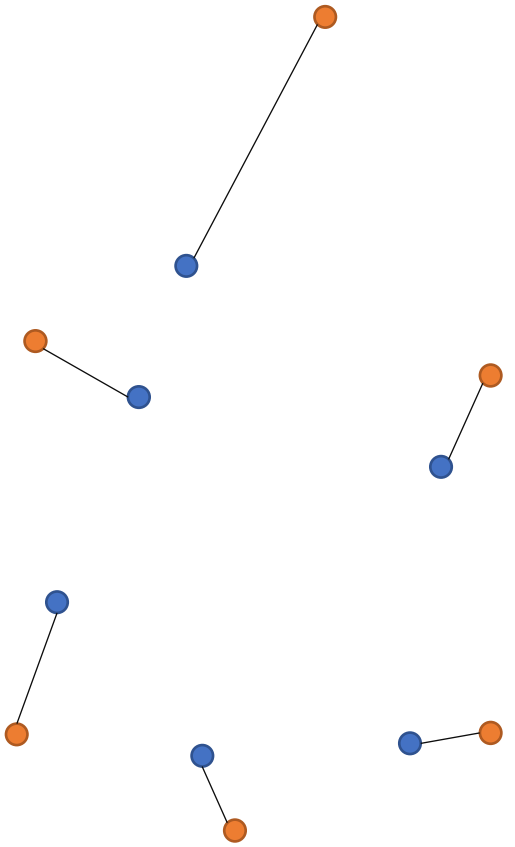
Worst case: Hausdorff distance (HD)

Average case: Chamfer distance (CD)

Optimal case: Earth Mover's distance (EMD)

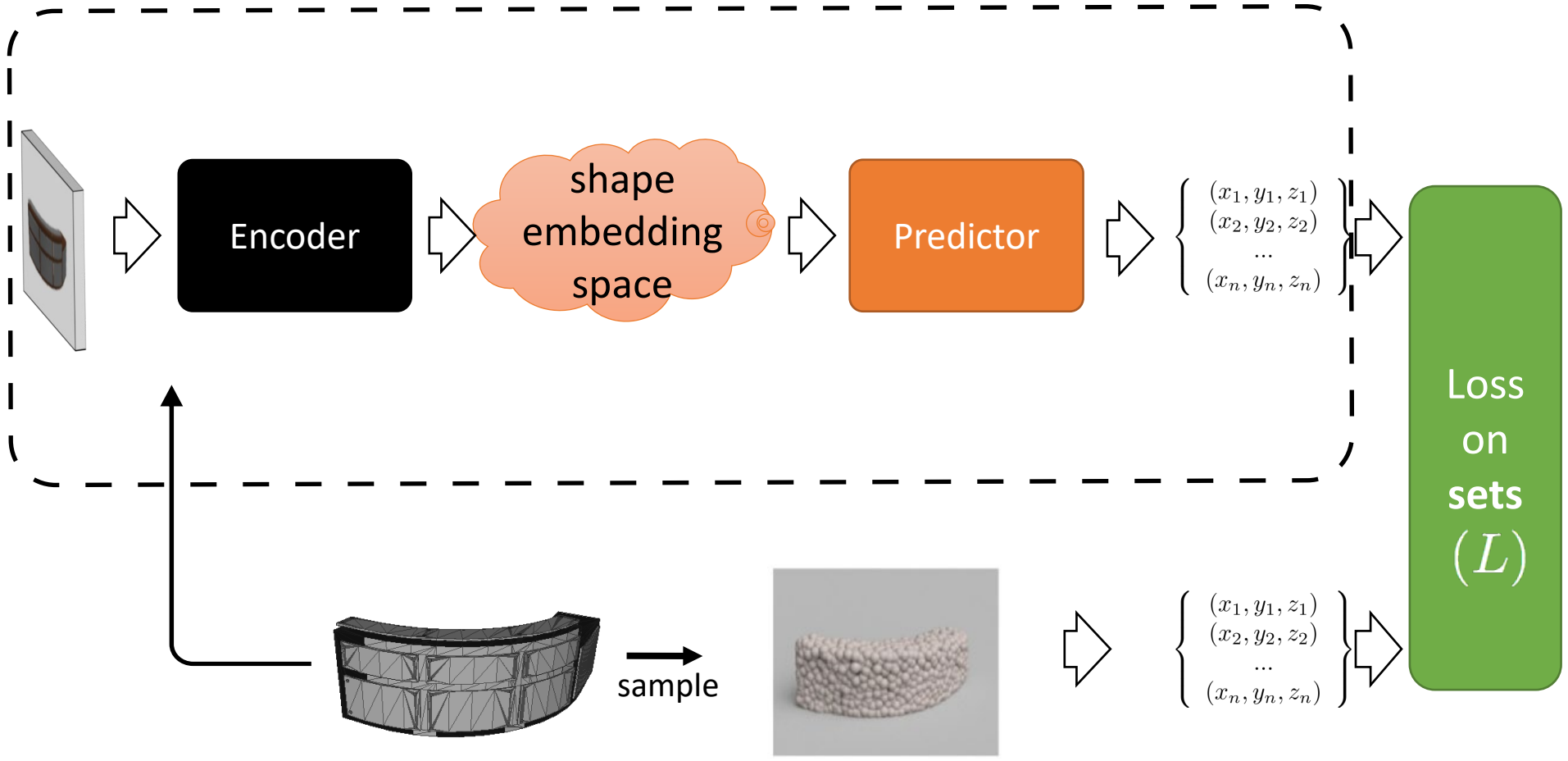
$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

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

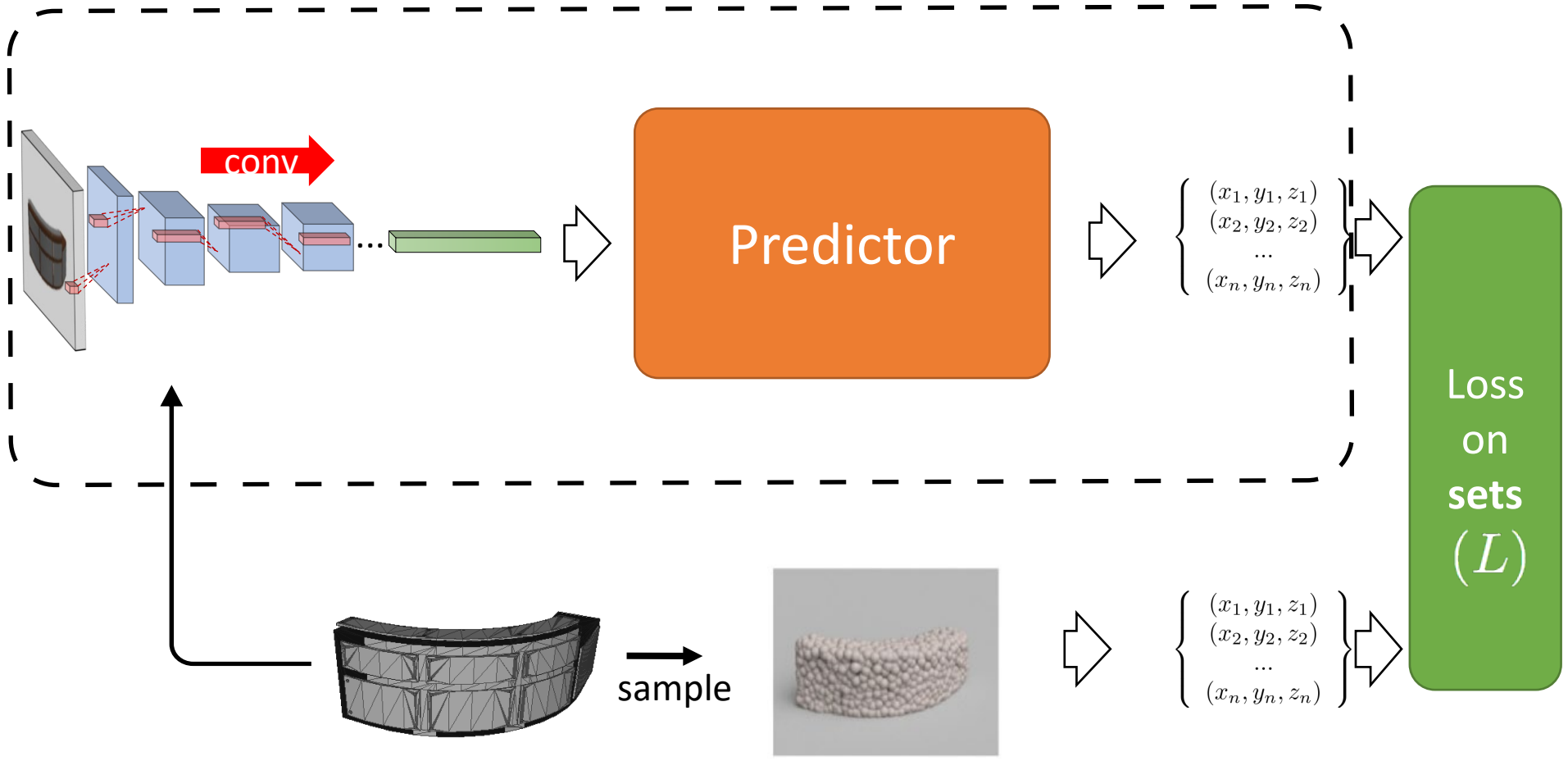


*Solves the optimal transportation (bipartite matching) problem!*

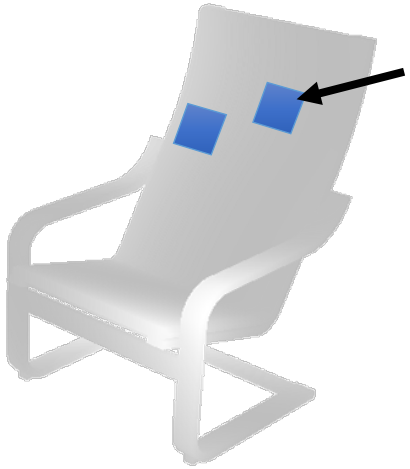
# End-to-End Learning

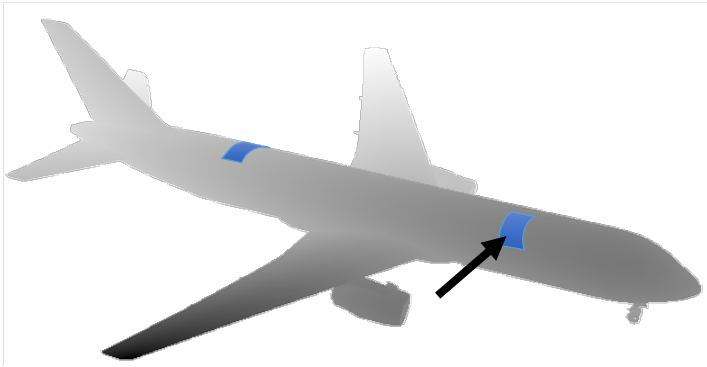


# End-to-End Learning



# Natural Statistics of Object Geometry

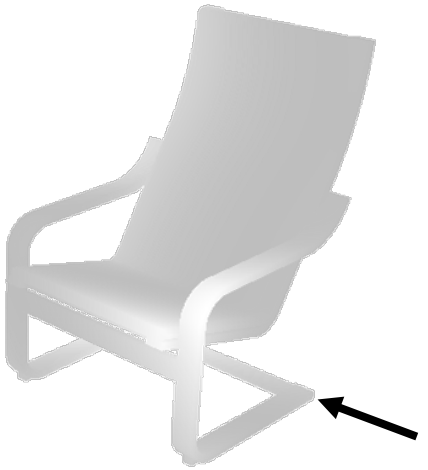




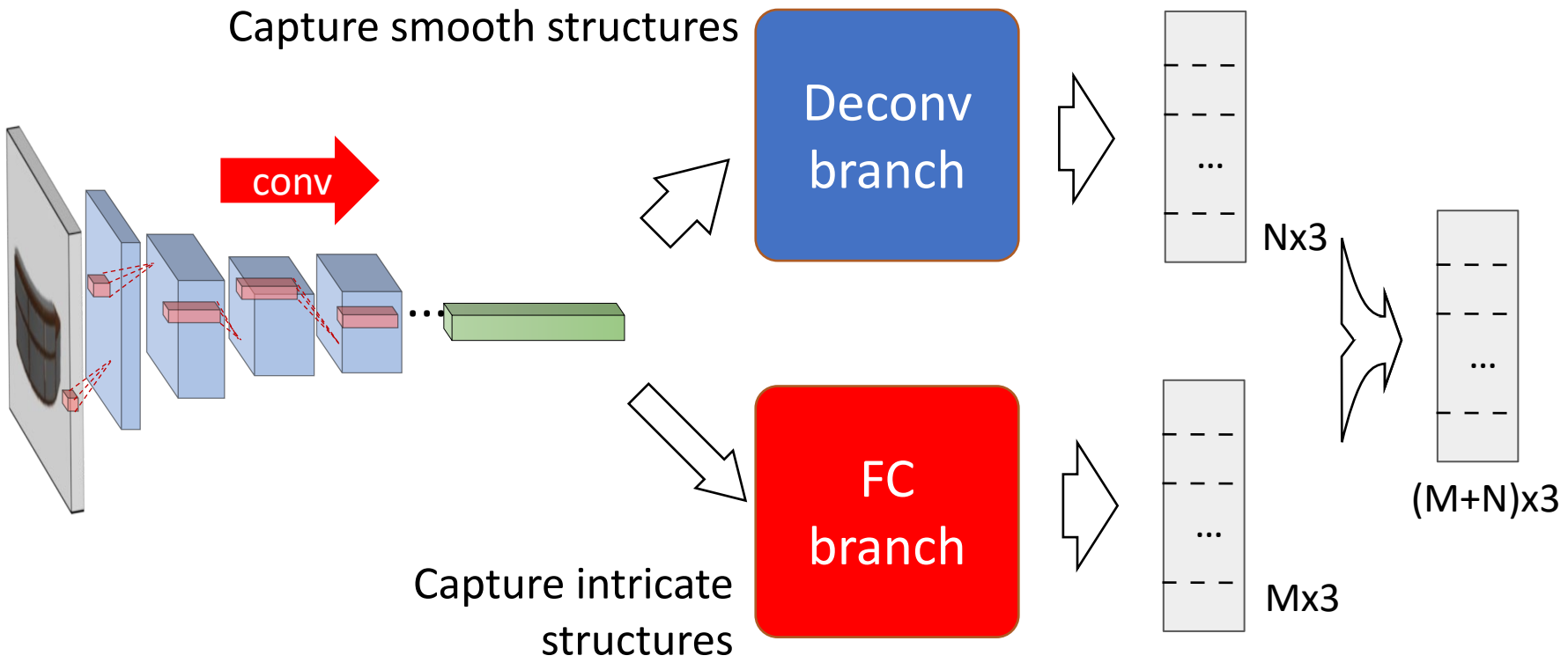
- Many local structures are common
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates

# Natural Statistics of Object Geometry

- Many local structures are common/shared
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates
- But also some intricate local structures
  - **some points have high variability neighborhoods**

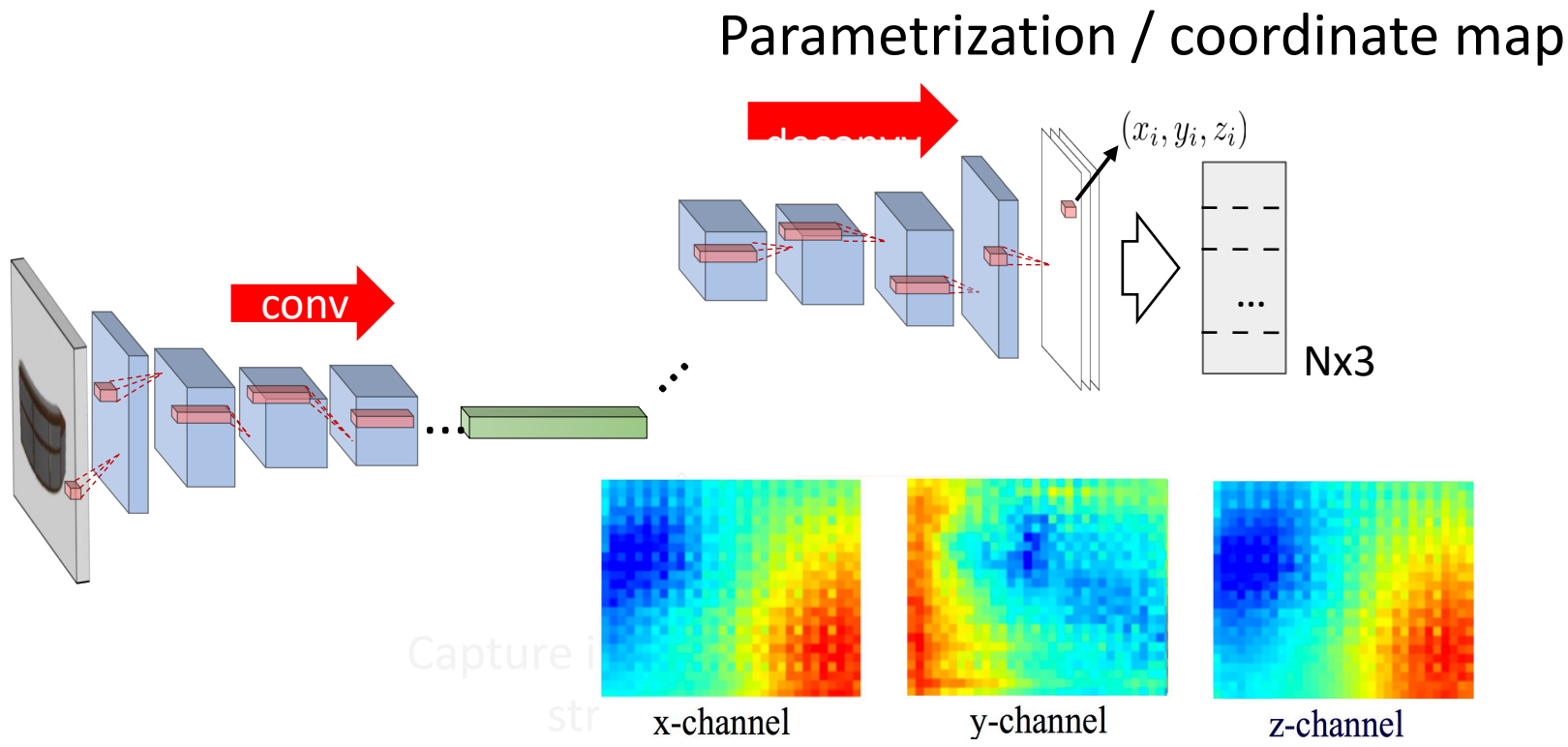


# Two-Branch Architecture



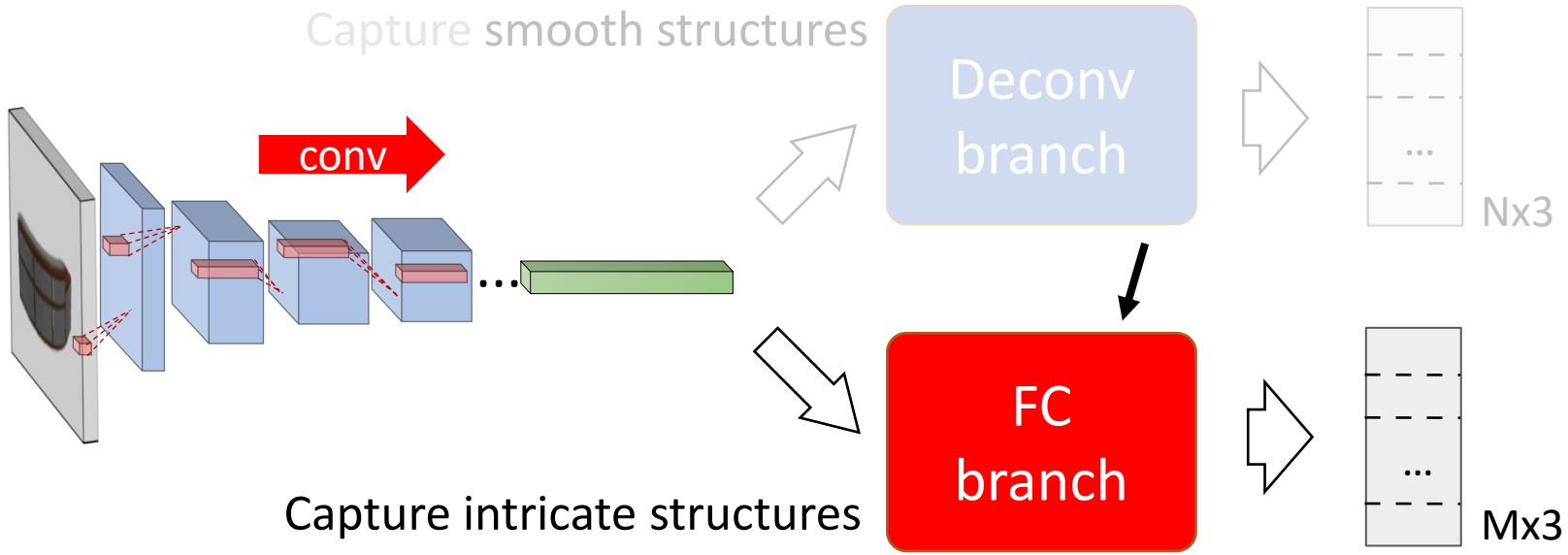

**Set union by array concatenation**

# Deconvolution Branch




- Deconvolution induces a smooth coordinate map
- Geometrically, learns a smooth parameterization

# Fully Connected Branch

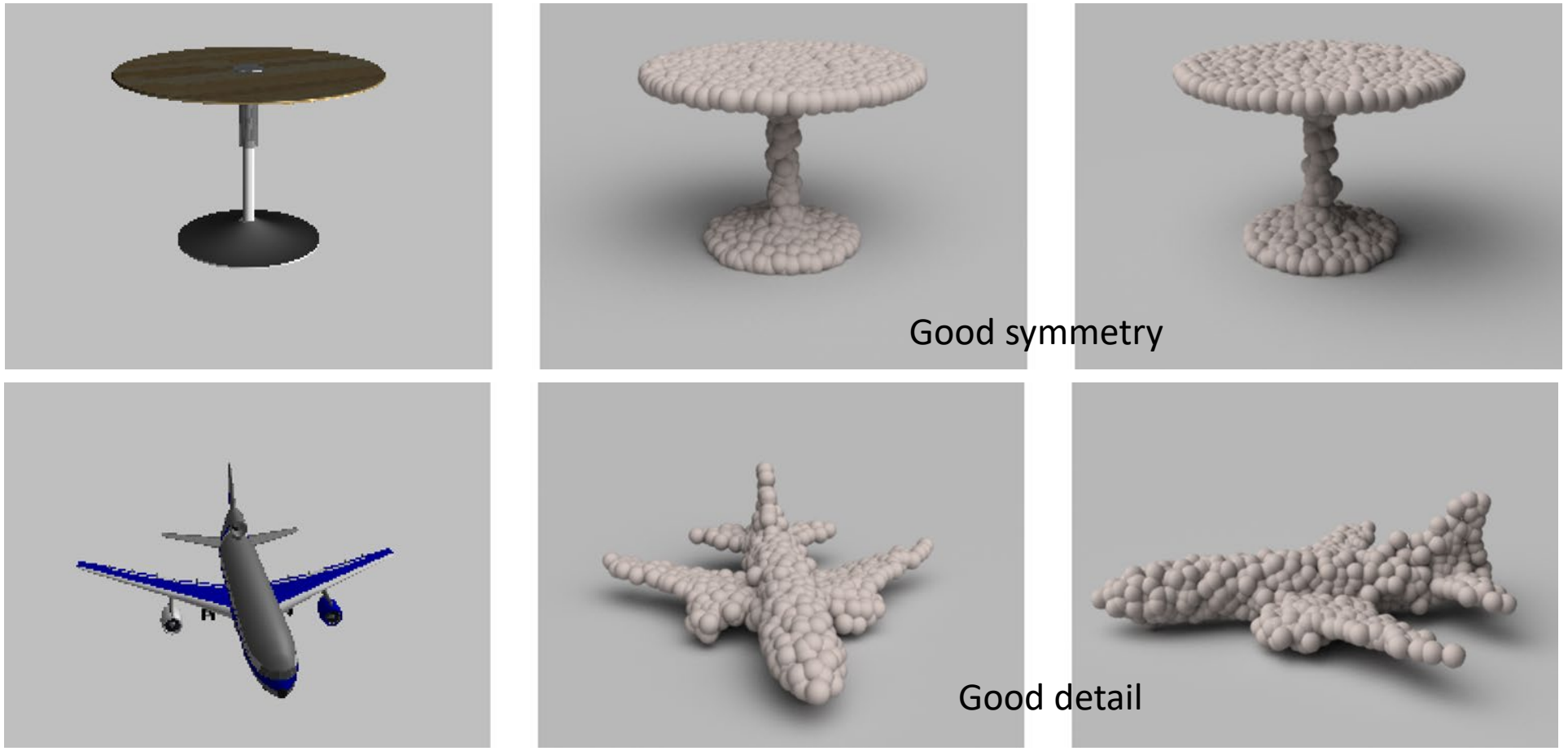


# The Two Branches

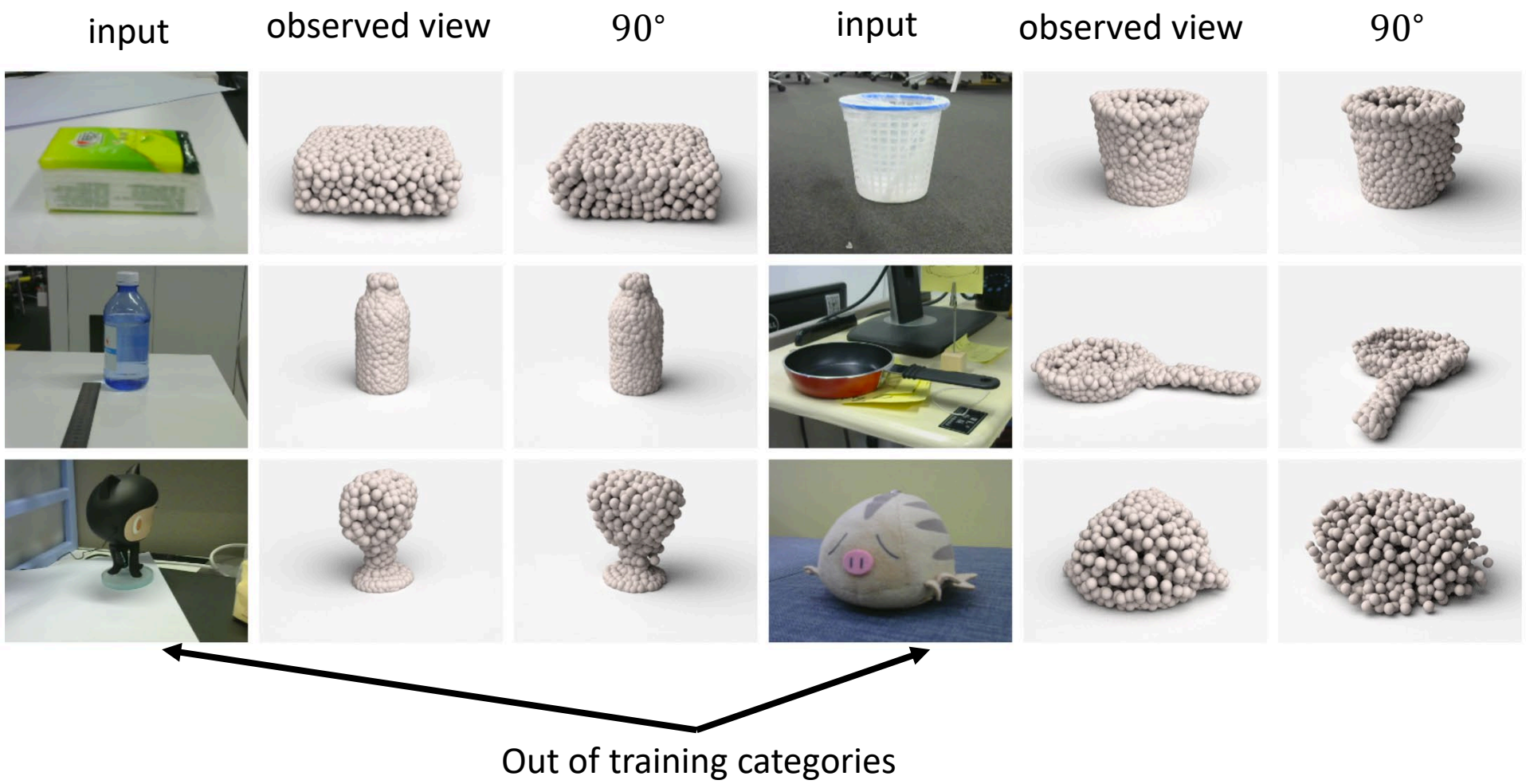
**blue**: deconv branch – **large, consistent, smooth** structures

**red**: fully-connected branch – **more intricate** structures

# Example Results

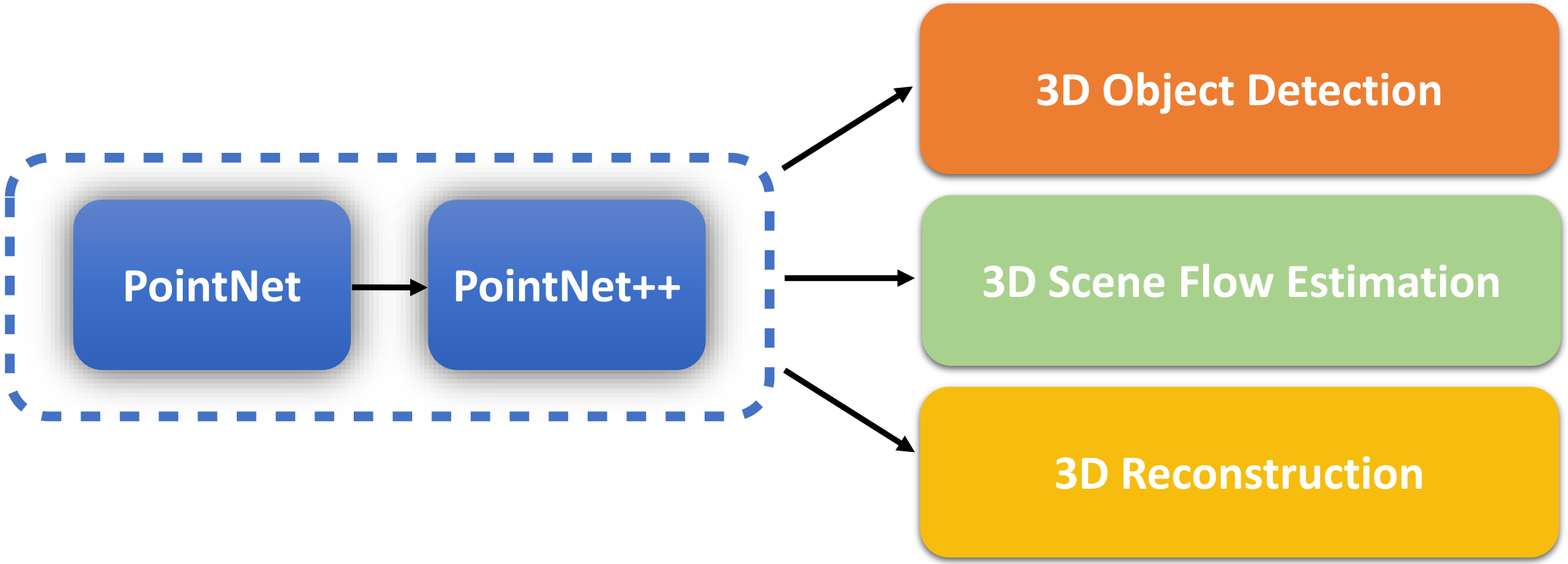


# From Real Images



# Conclusions: Real-World 3D Understanding

- **Novel architectures for deep learning on point clouds** – PointNet and PointNet++, respecting invariances, light-weight and robust to data corruption, a unified framework for various tasks.
- **Successful applications in 3D scene understanding**.



# That’s All