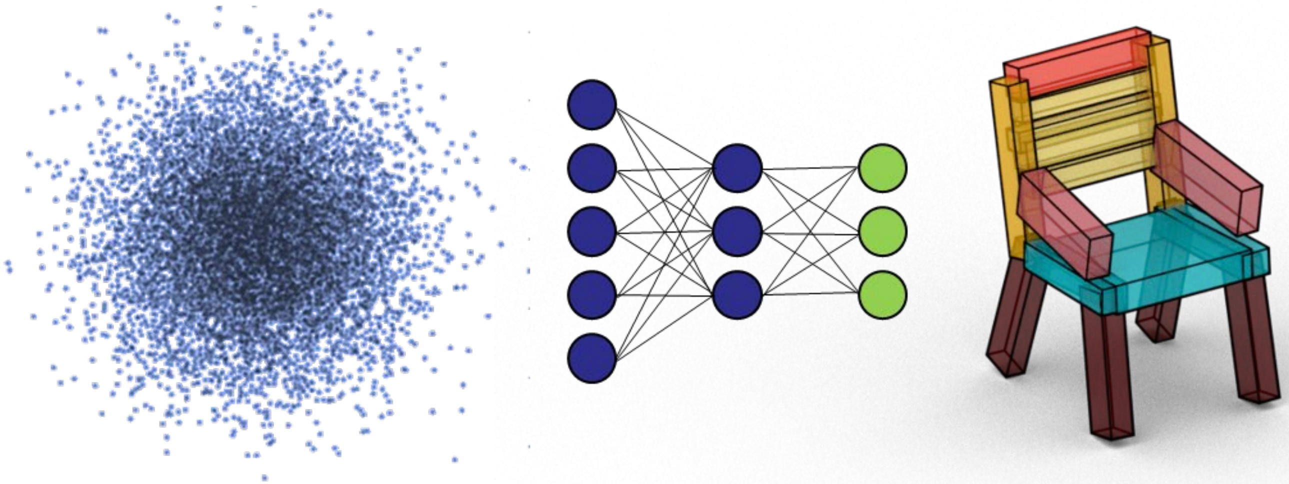
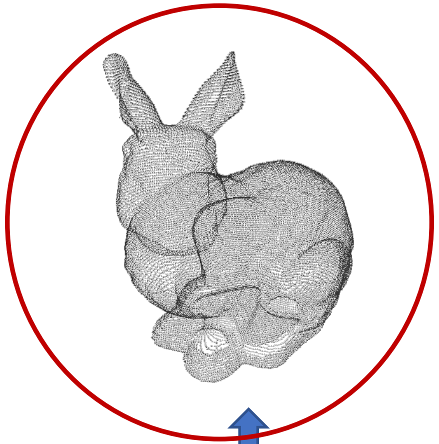
CS348n: Neural Representations and Generative Models for 3D Geometry

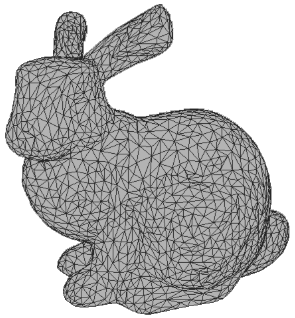


Leonidas Guibas
Computer Science Department
Stanford University

Leonidas Guibas Laboratory
Geometric Computing

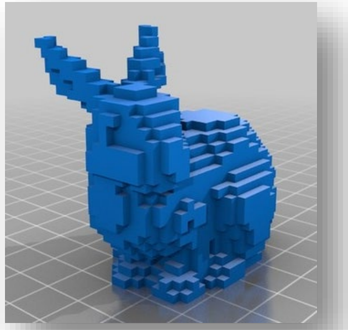
In 3D, There is Representation Diversity





Point Cloud

Mesh

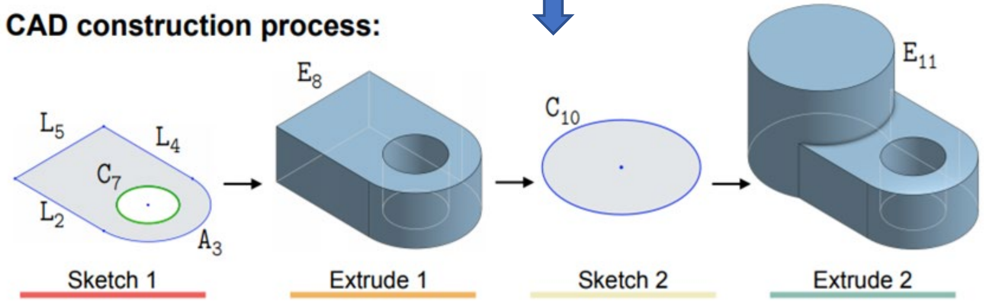


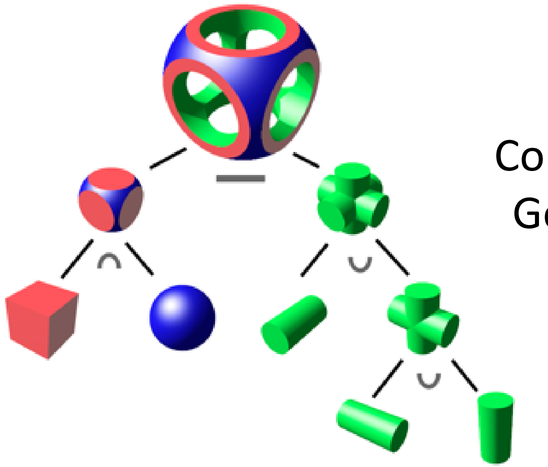
Voxels

Multiple View Images
RGB(D)

These are irregular representations – and the ones most commonly used in 3D apps

Sketch-Extrude





Constructive Solid Geometry (CSG)

Summer School on Geometry Processing

Summer Geometry Initiative

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

SGI 2022

The Summer Geometry Initiative (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 (online) in 2022, 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

Prof Justin Solomon, MIT

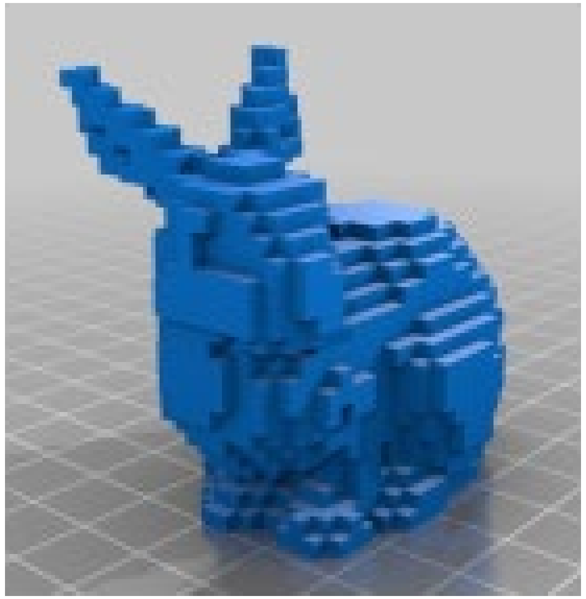
https://sgi.mit.edu/

Next week

- Mon, Jan 17, MLK holiday
- Wed, Jan 19, Zoom class
- Fri, Jan 21, 1:00-3:00 pm, in person extended office hours

Last Time: Volumetric Representations

Volumetric Representation: 3D Geometry



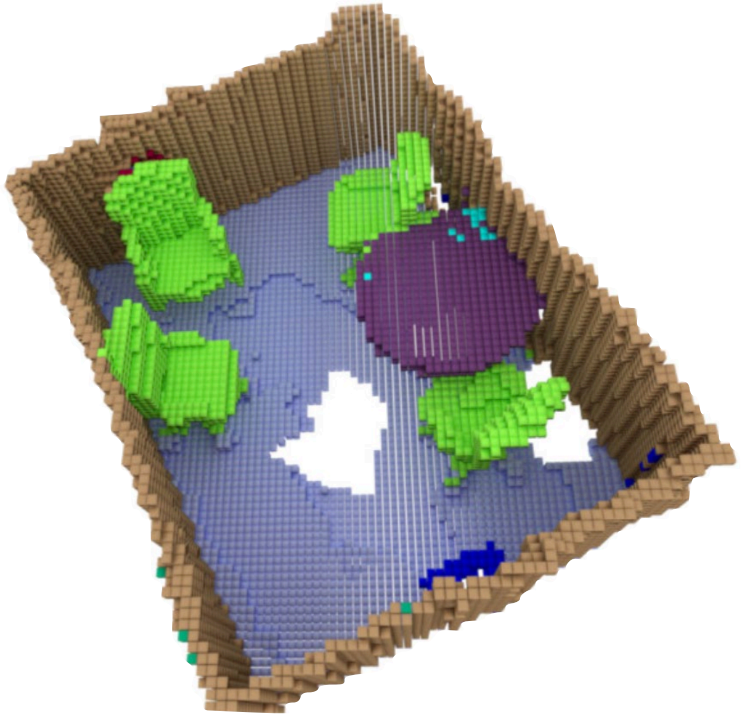
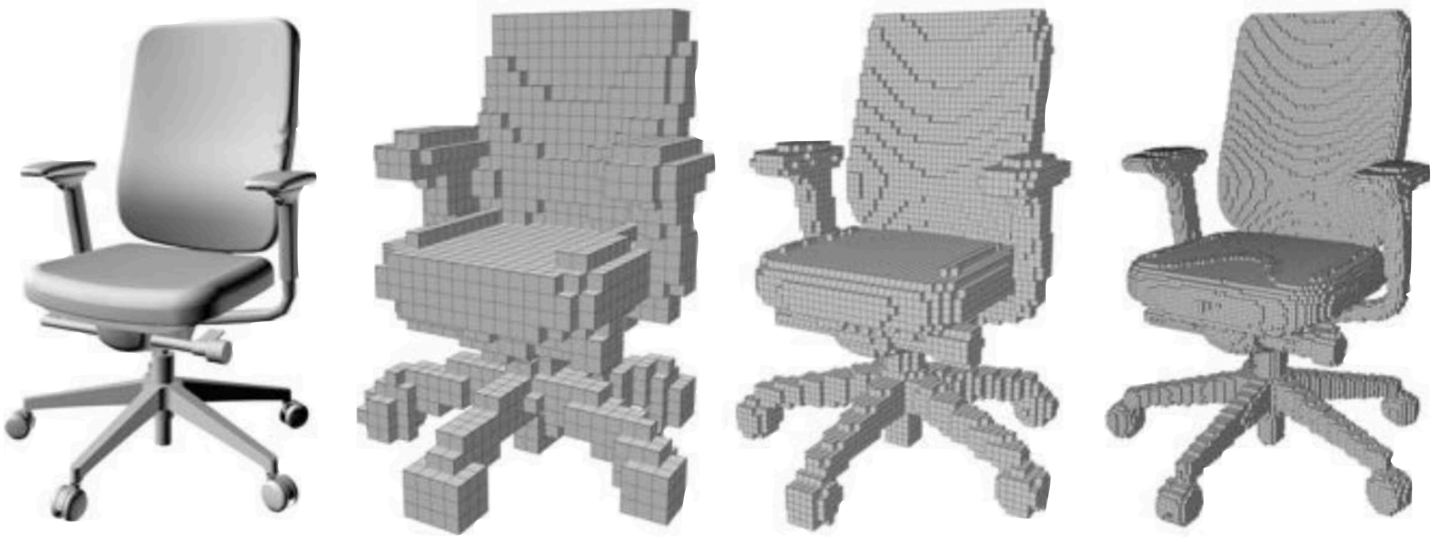


Image Credits: Scannet

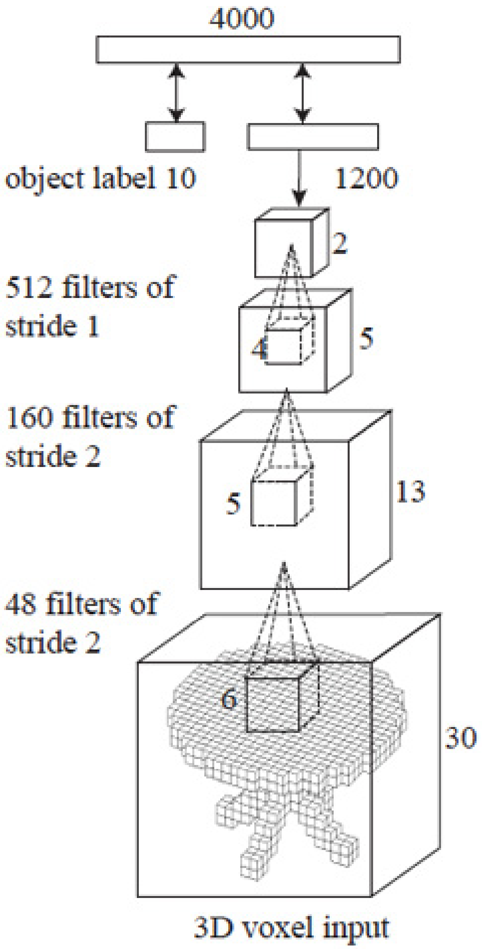
Volumetric Representation



		32	64	128
$\frac{\#occupied\ grid}{\#total\ grid}$	Occupancy:	10.41%	5.09%	2.41%
	Resolution:	32	64	128

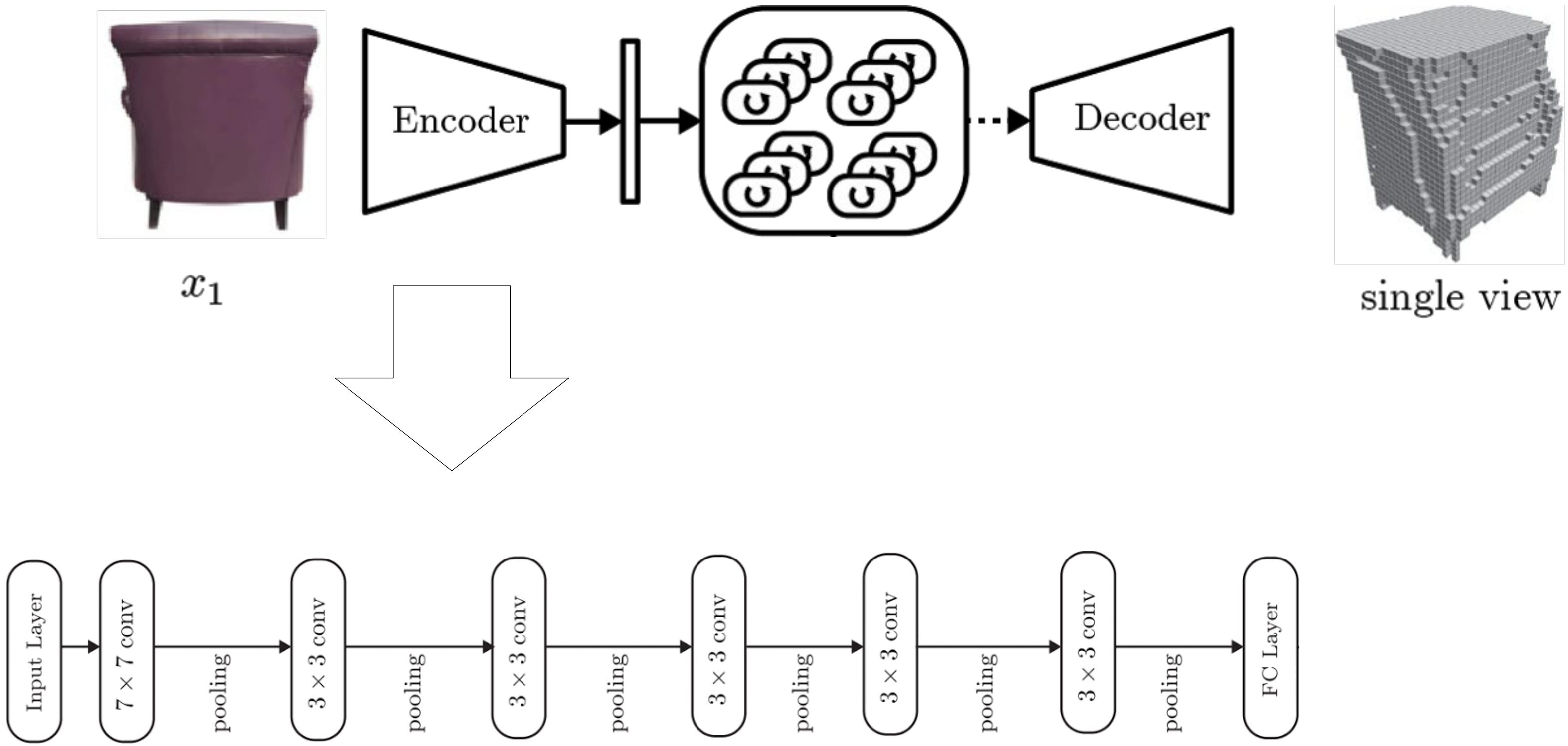
Resolution *N* O(*N*^3)

3D Voxel CNNs: Classification



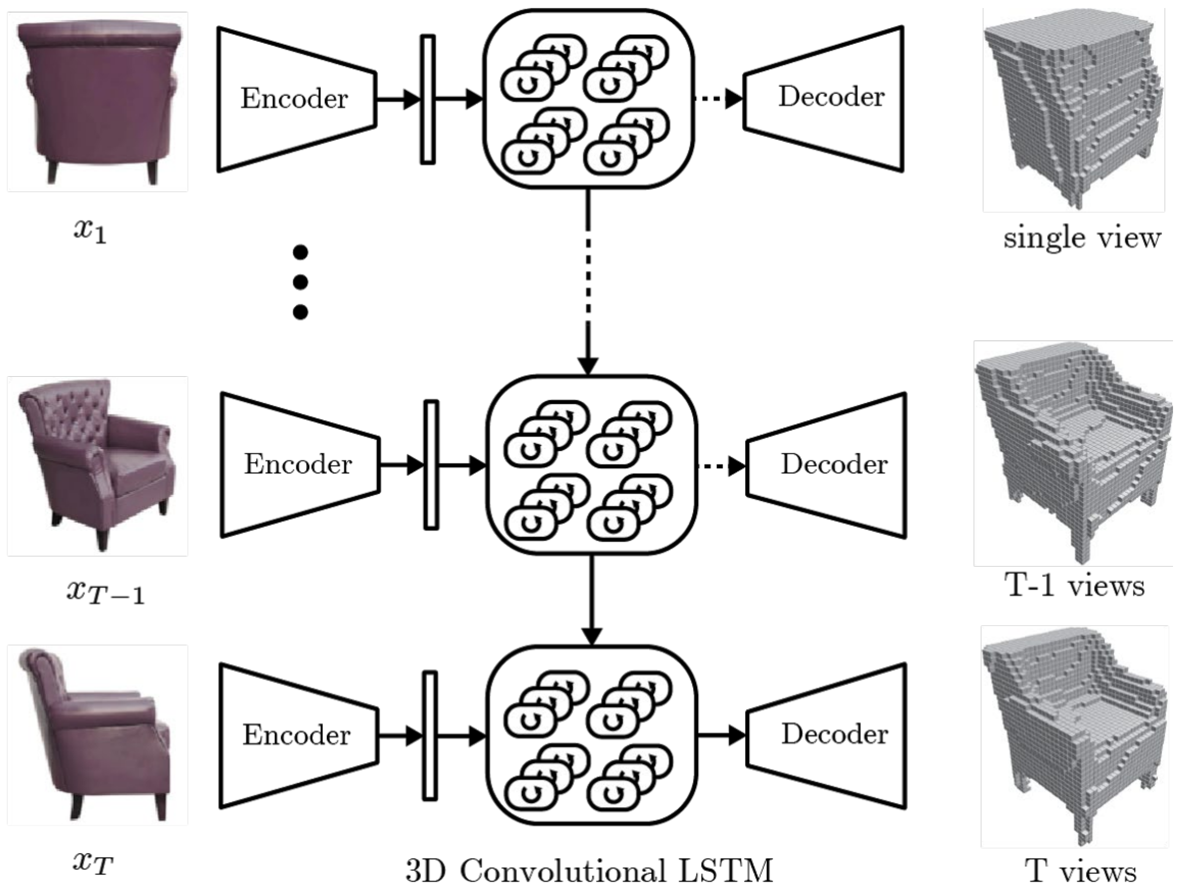
Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang and J. Xiao, **"3D ShapeNets: A Deep Representation for Volumetric Shape Modeling",** *CVPR2015*

3D Voxel CNNs: Reconstruction



3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction
Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese
ECCV 2016

3D Voxel CNNs: Reconstruction

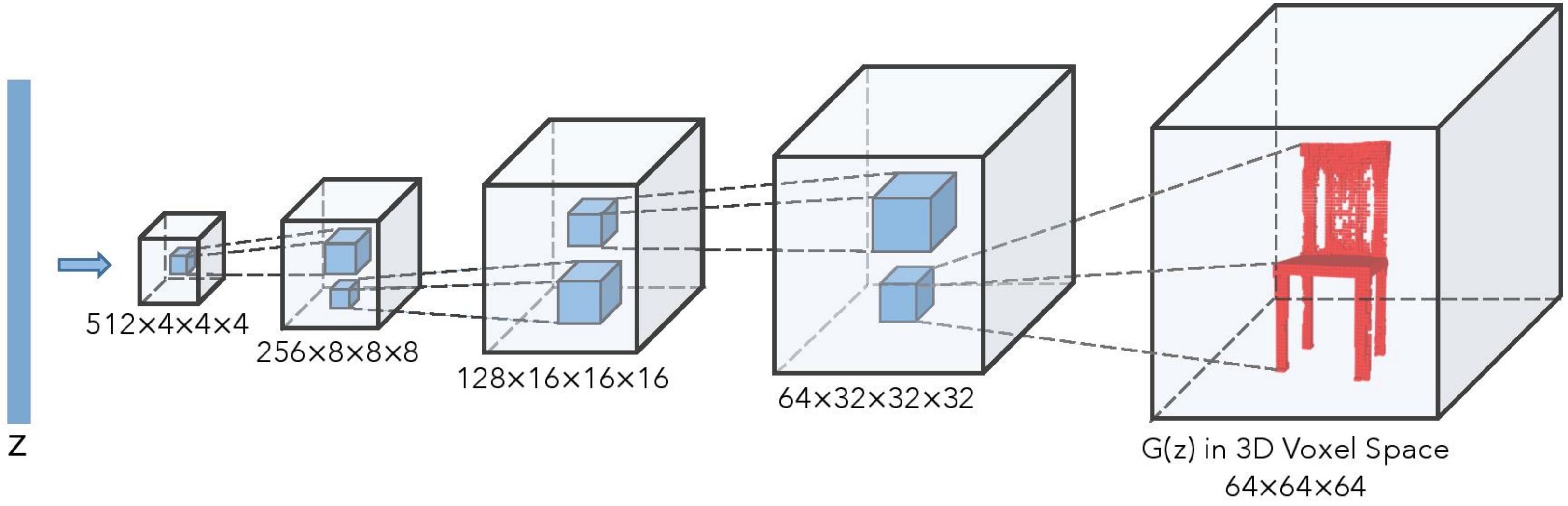


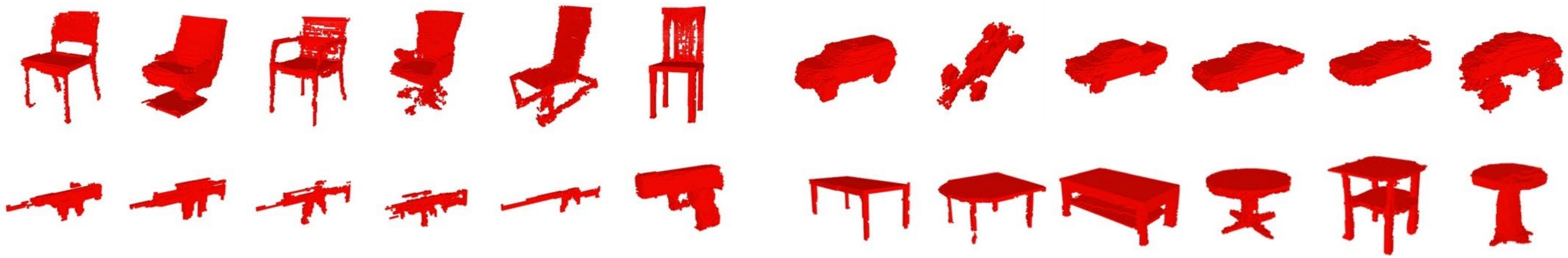
3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction
Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese
ECCV 2016

3D Voxel CNNs: Reconstruction

3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction
Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese
ECCV 2016

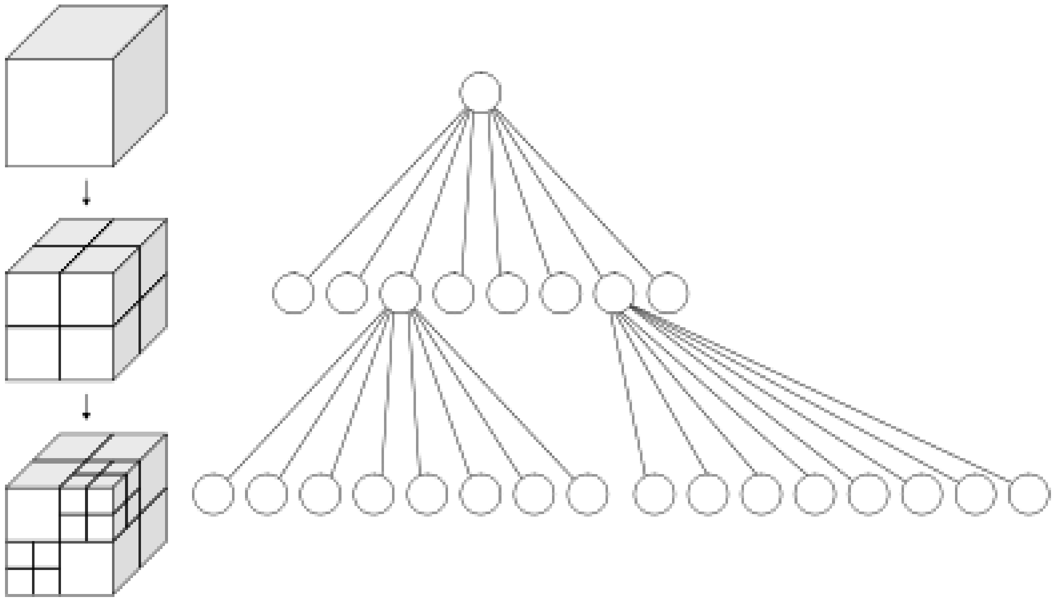
3D Voxel CNNs: Generation



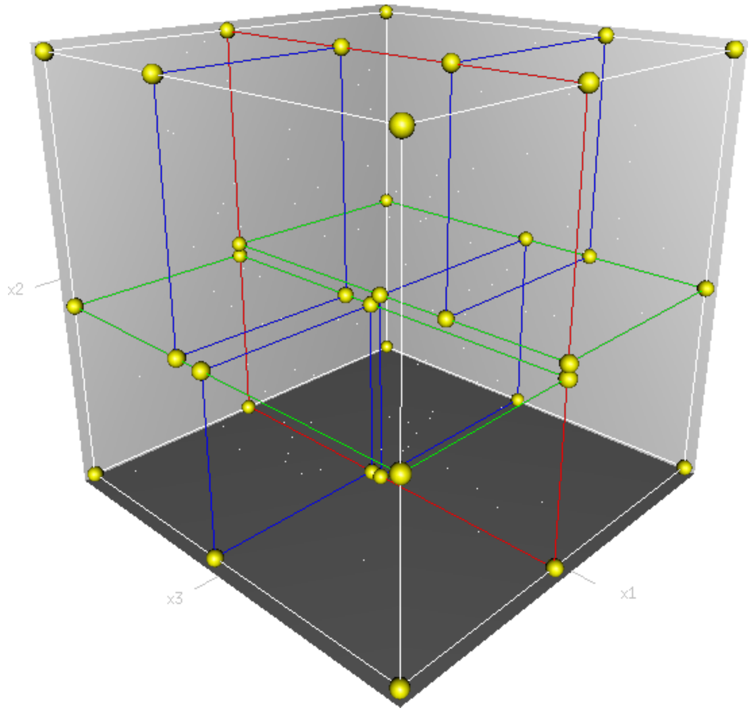


Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
Jiajun Wu*, Chengkai Zhang*, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum
NeurIPS2016

Hierarchical Volumetric Representation



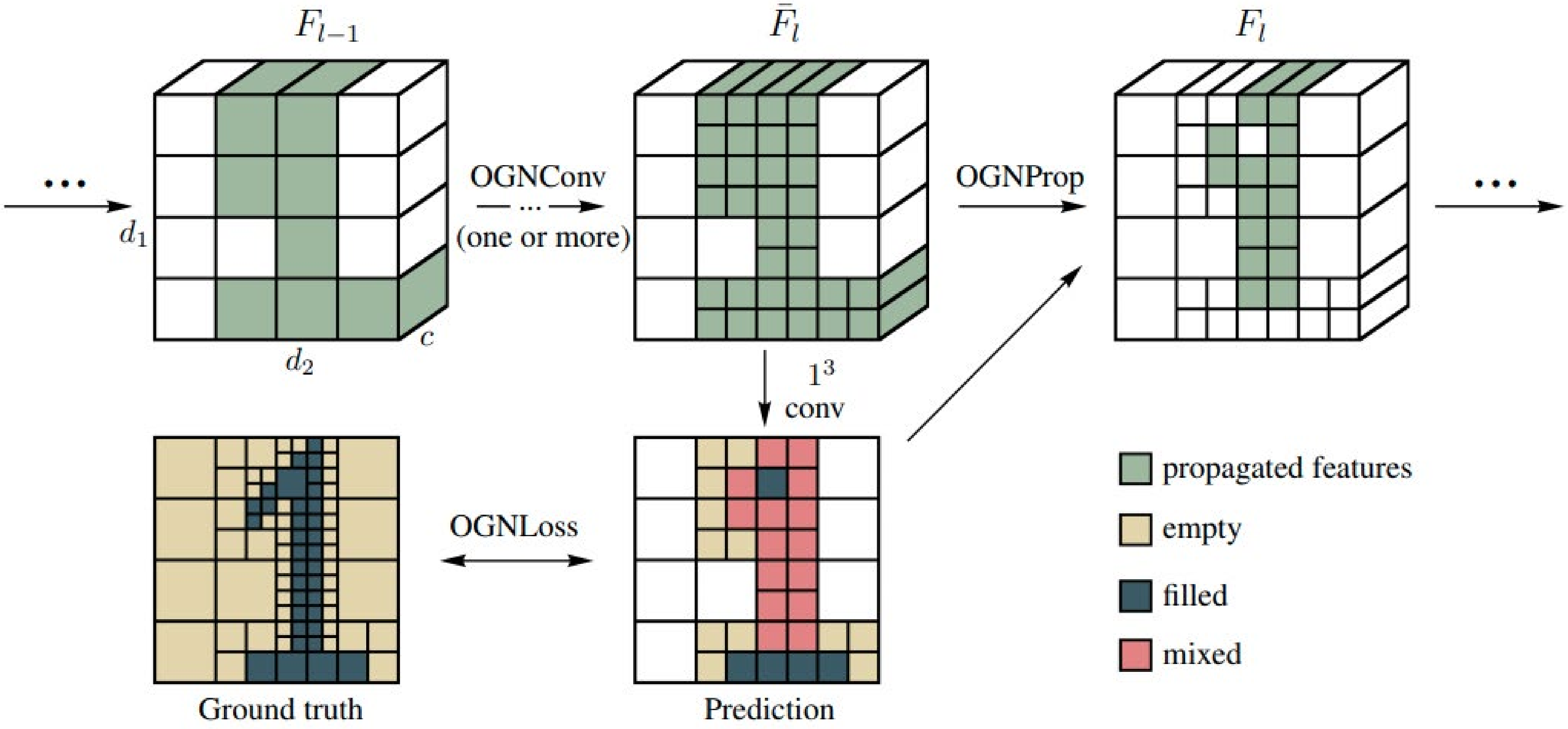
Octree



3D KD-Tree

Image Credits: Wikipedia

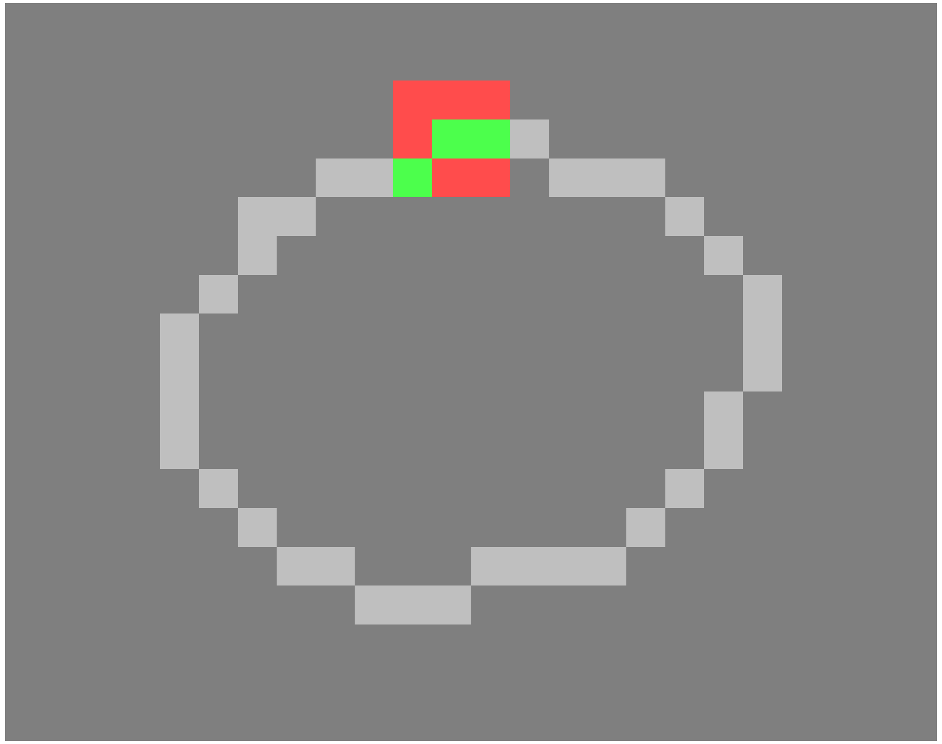
Hier 3D Voxel NN: Generation



Maxim Tatarchenko, Alexey Dosovitskiy, Thomas Brox

"Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs"
ICCV, 2017

3D Sparse ConvNets

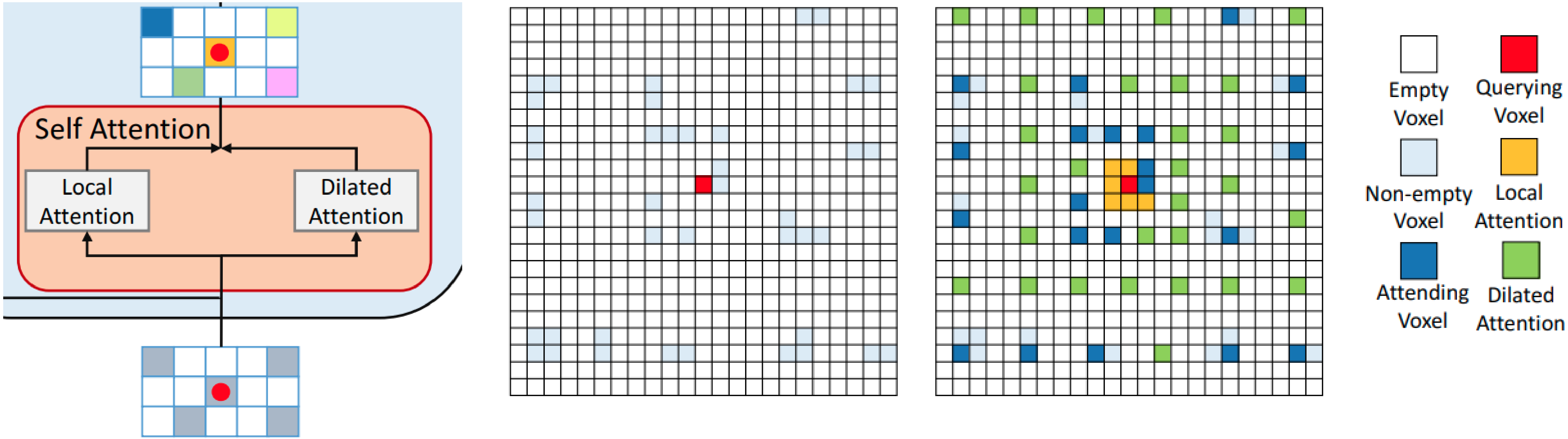


Submanifold Sparse Convolution (SSC)

- carefully engineered that no computational overhead at the empty cells, using a ***hash-table*** and a ***rule-book***
- only computed when the kernel ***center*** is over an ***occupied*** cell

3D Semantic Segmentation with Submanifold Sparse Convolutional Networks
***Benjamin* Graham, Martin Engelcke, Laurens van der Maaten,** CVPR, 2018

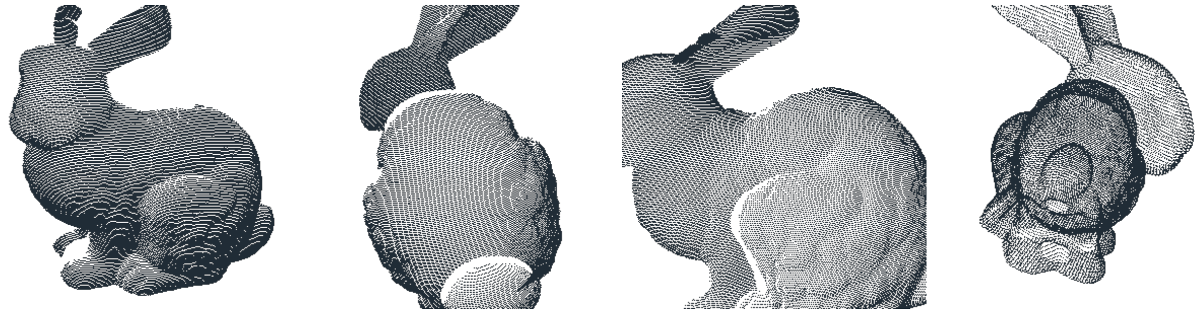
VoTr: Voxel Transformer



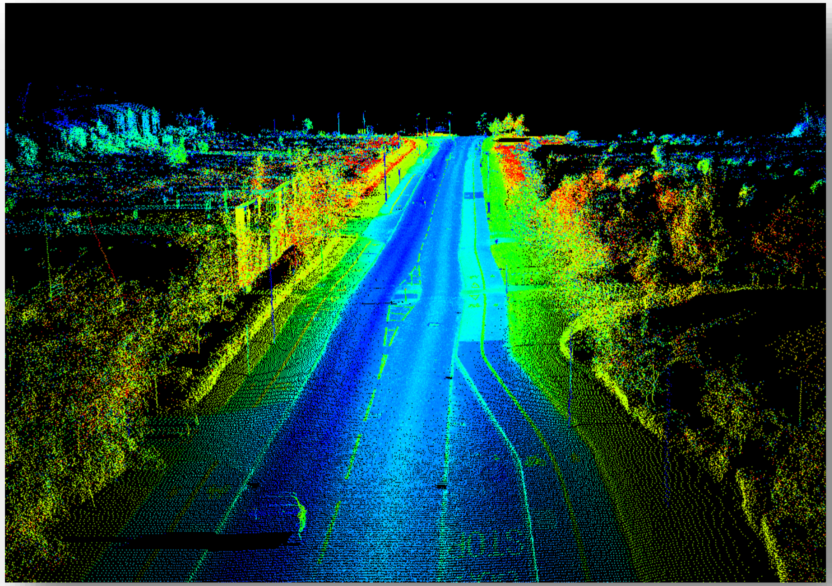
using efficient Sparse Operations for Querying, Retrieval, Convolution, Multiplication, etc.

Voxel Transformer for 3D Object Detection
Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, ICCV, 2021

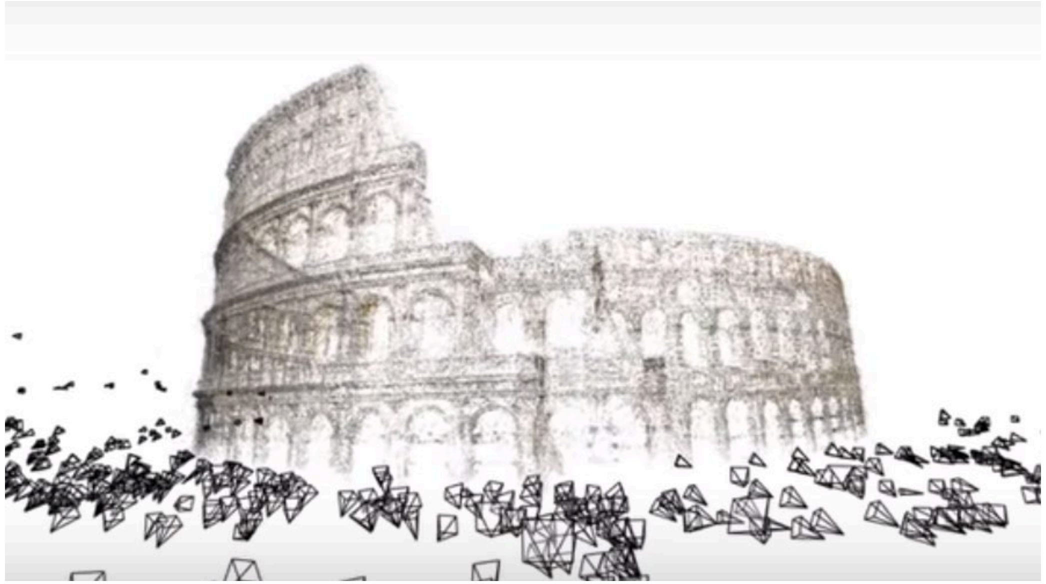
Point Clouds



3D Point Clouds from Many Sensors



Lidar point clouds (LizardTech)



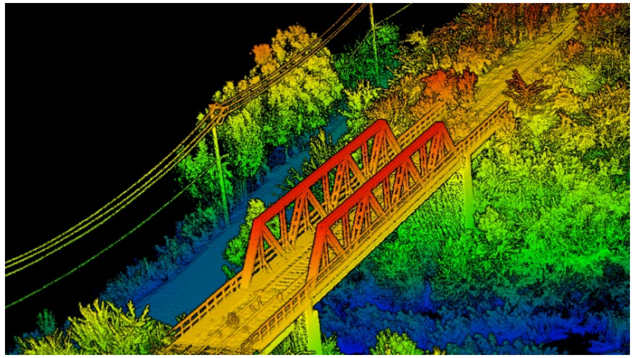
Structure from motion (Microsoft)

Depth camera (Intel, Microsoft, Google)

3D Point Cloud Data

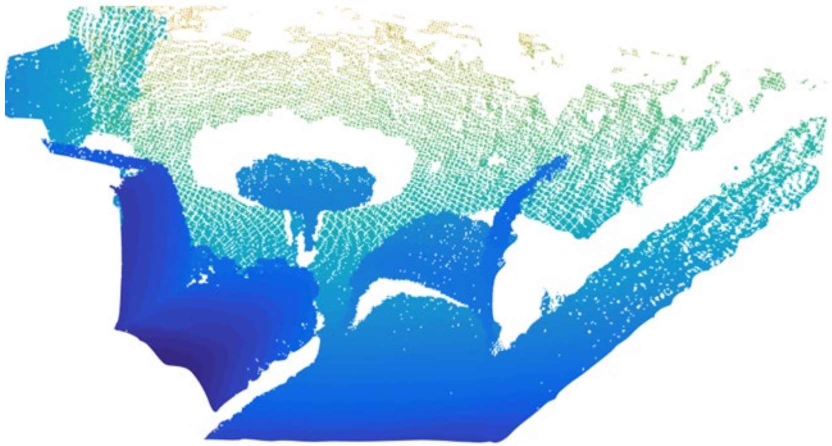
- Close to raw sensor data
- Representationally simple
- Irregular neighborhoods
- Variable density

LiDAR

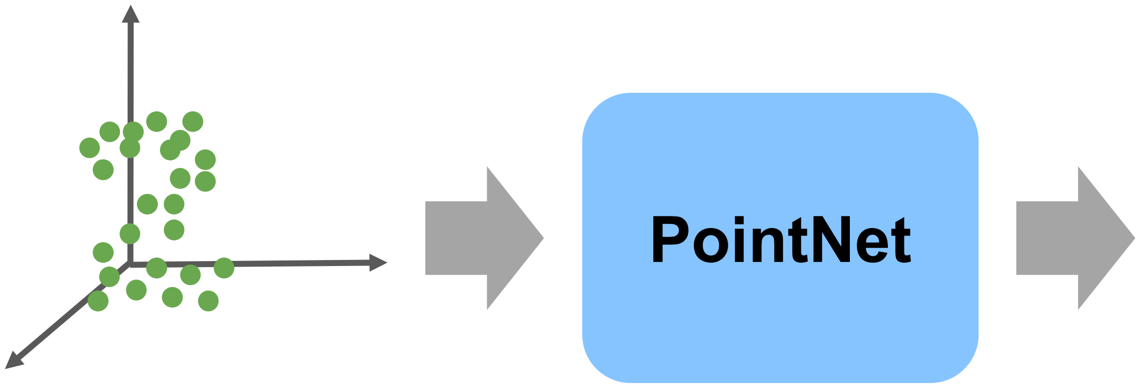


Depth Sensor

Point Cloud



Deep Nets for PCs: PointNet and PointNet++



Object Classification

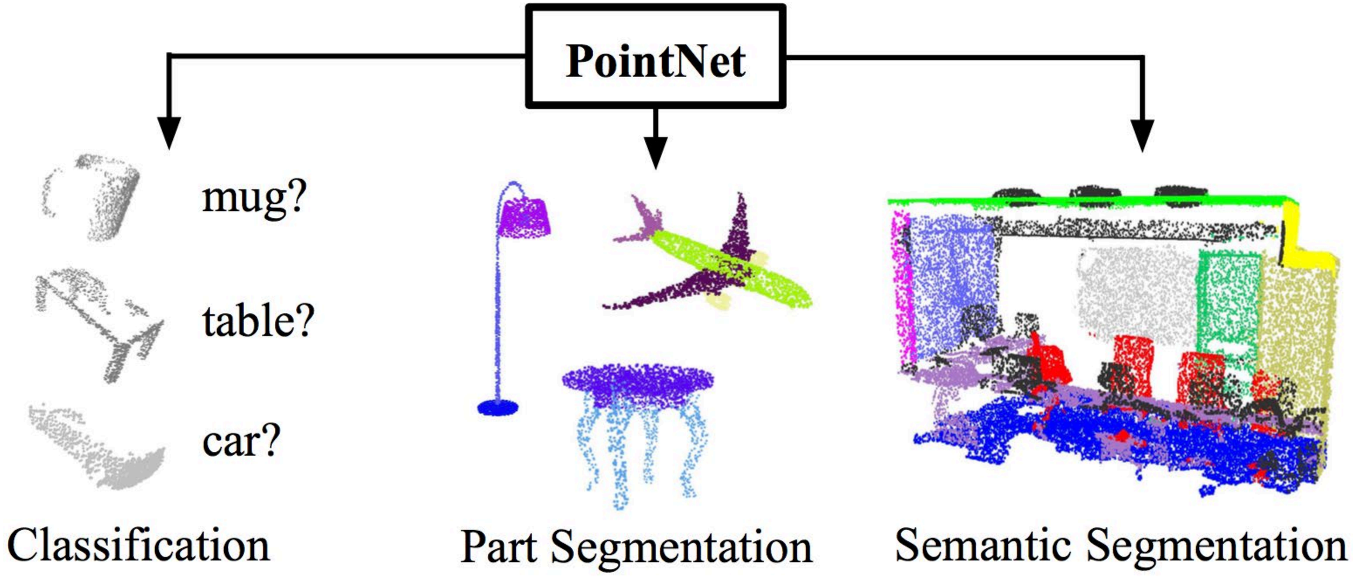
Object Part Segmentation

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:

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

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

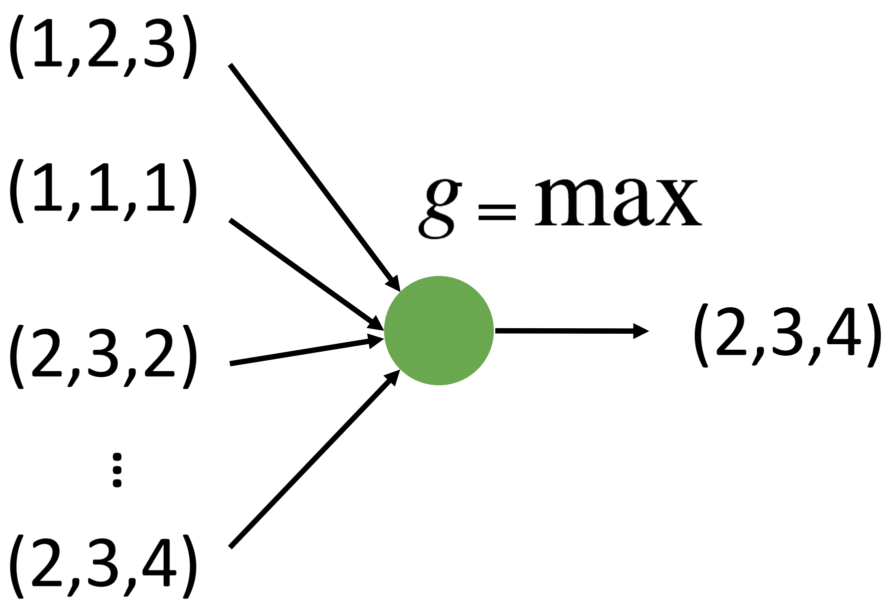
...

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

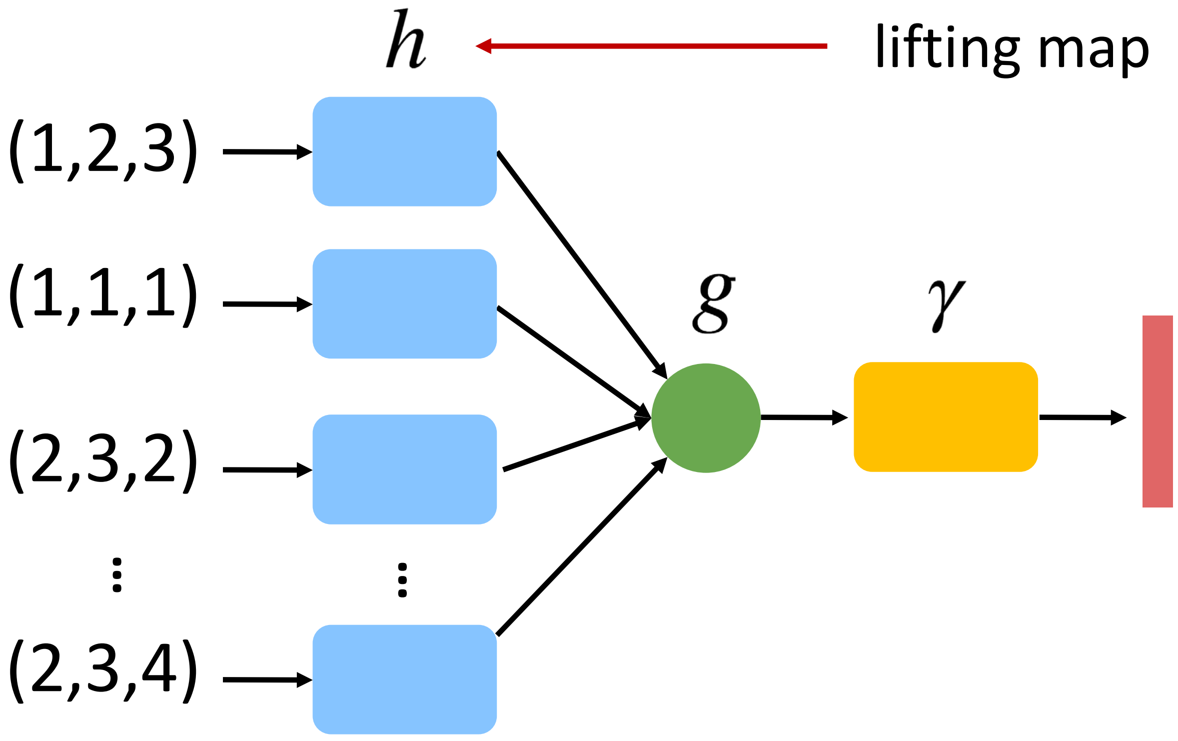
Just discovers simple extreme/aggregate properties of the geometry.



Construct Symmetric Functions by Neural Networks

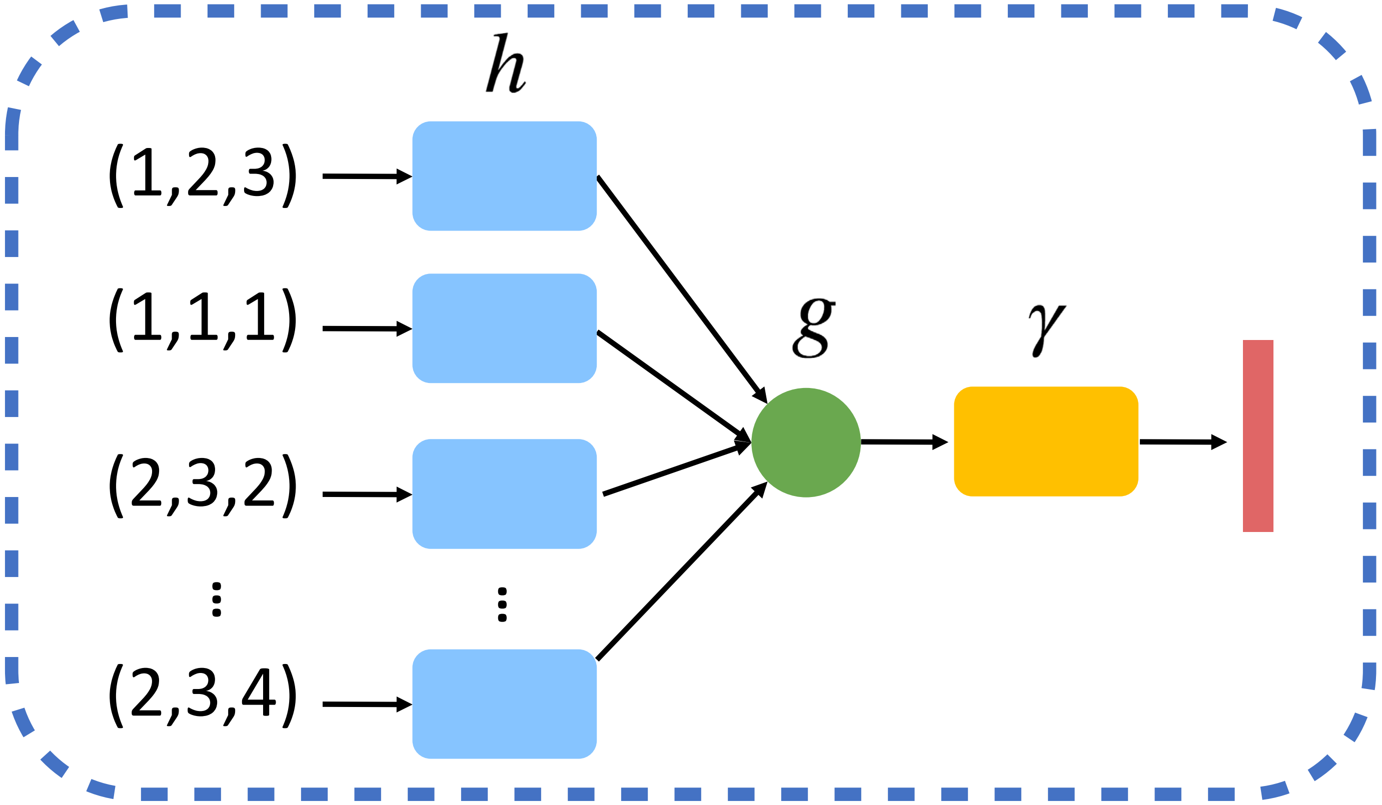
Embed points in a high-dim space before aggregation.

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



Construct Symmetric Functions by Neural Networks

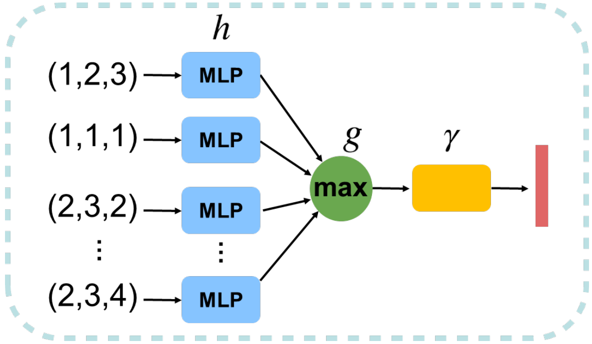
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



h lifts points to a high-dimensional space

PointNet (vanilla)

Symmetric Functions: Polynomials



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

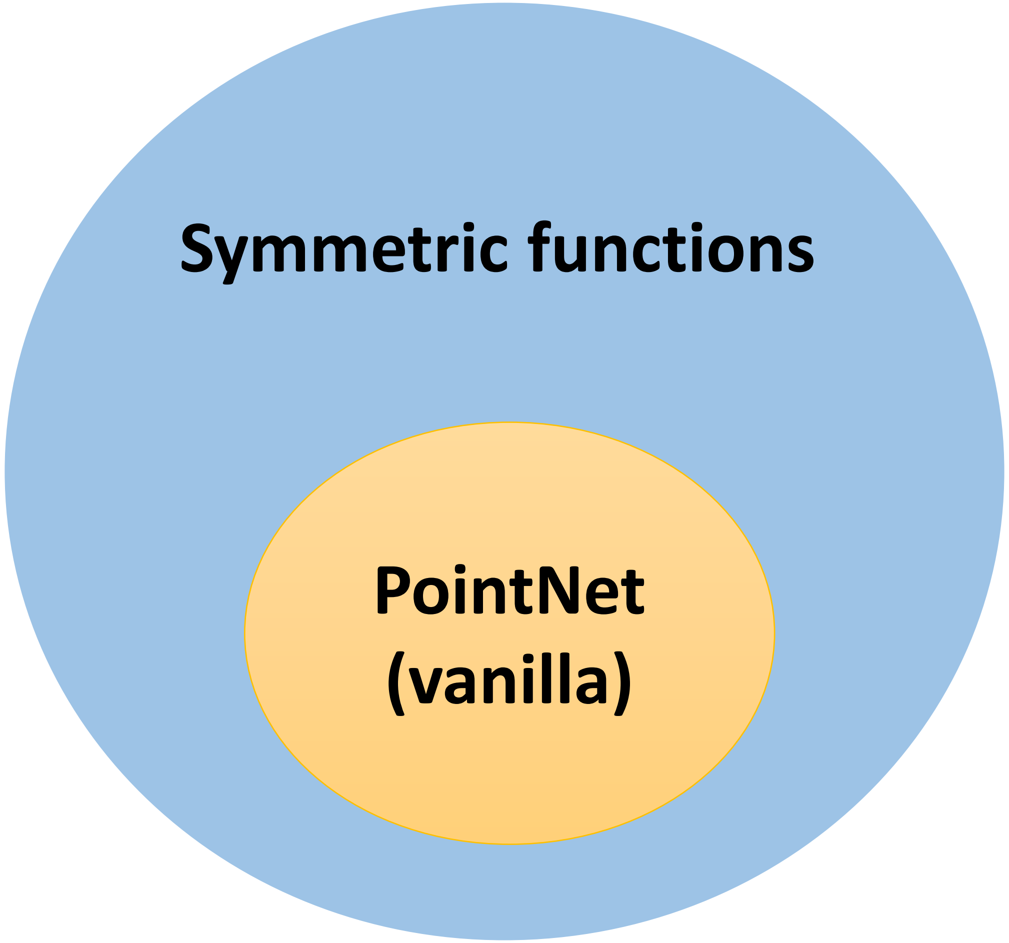
- In fact, any symmetric polynomial in the x_i can be expressed as a polynomial in sums of the form

$$\sum_i x_i^k$$

and can be computed by

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

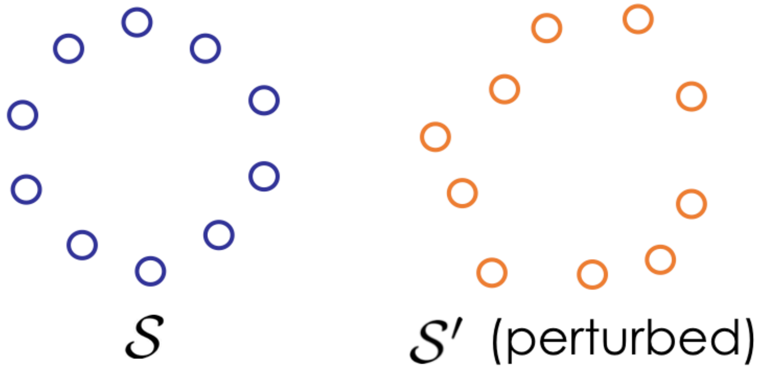
What Symmetric Functions Can Be Constructed By PointNet?



PointNet as a Universal Approximation to Set Functions

Hausdorff continuous:

$f: 2^{x} \rightarrow \mathbb{R}$ 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 $f: 2^{x} \rightarrow \mathbb{R}$ can be arbitrarily approximated by PointNet.

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

$S \subseteq R^d$

PointNet (vanilla)

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

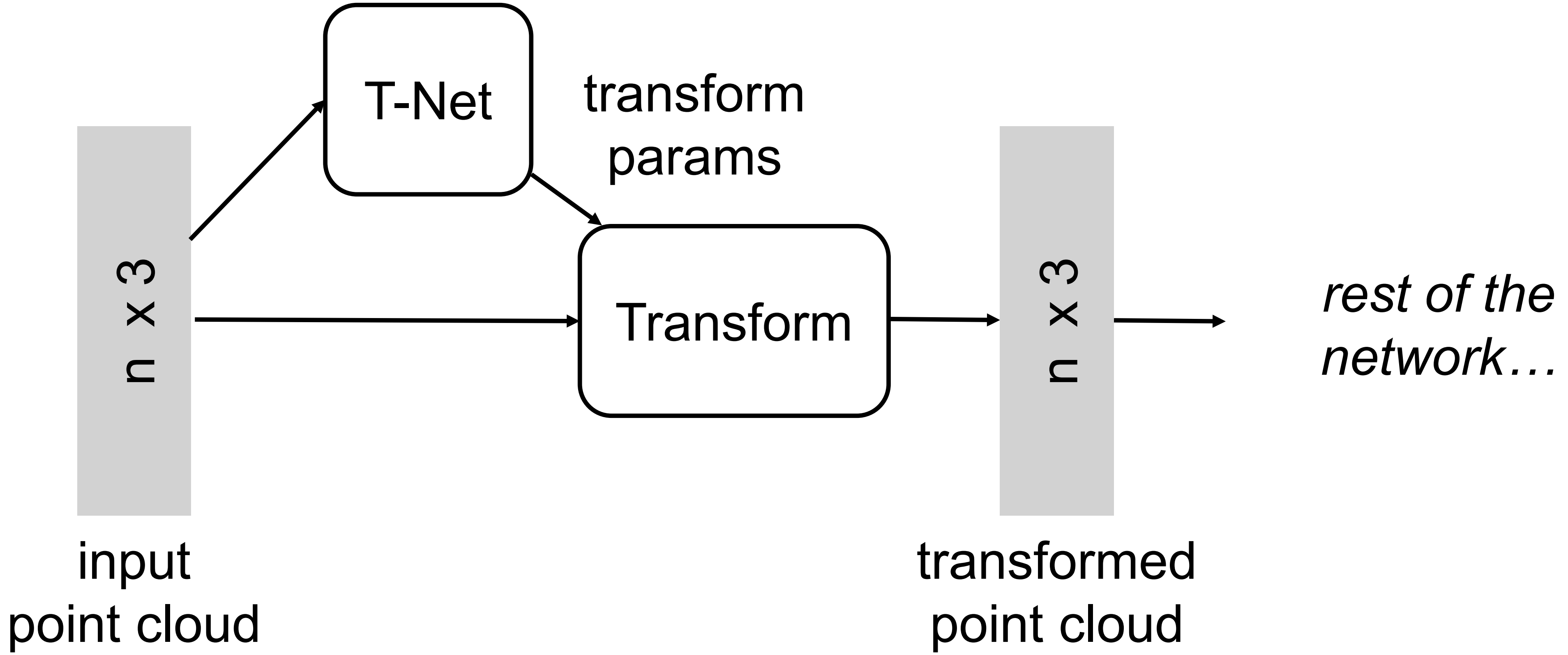
Point cloud rigid motions should not alter classification results

Sampling Invariance

Output a function of the underlying geometry and not the sampling

Input Alignment by Transformer Network

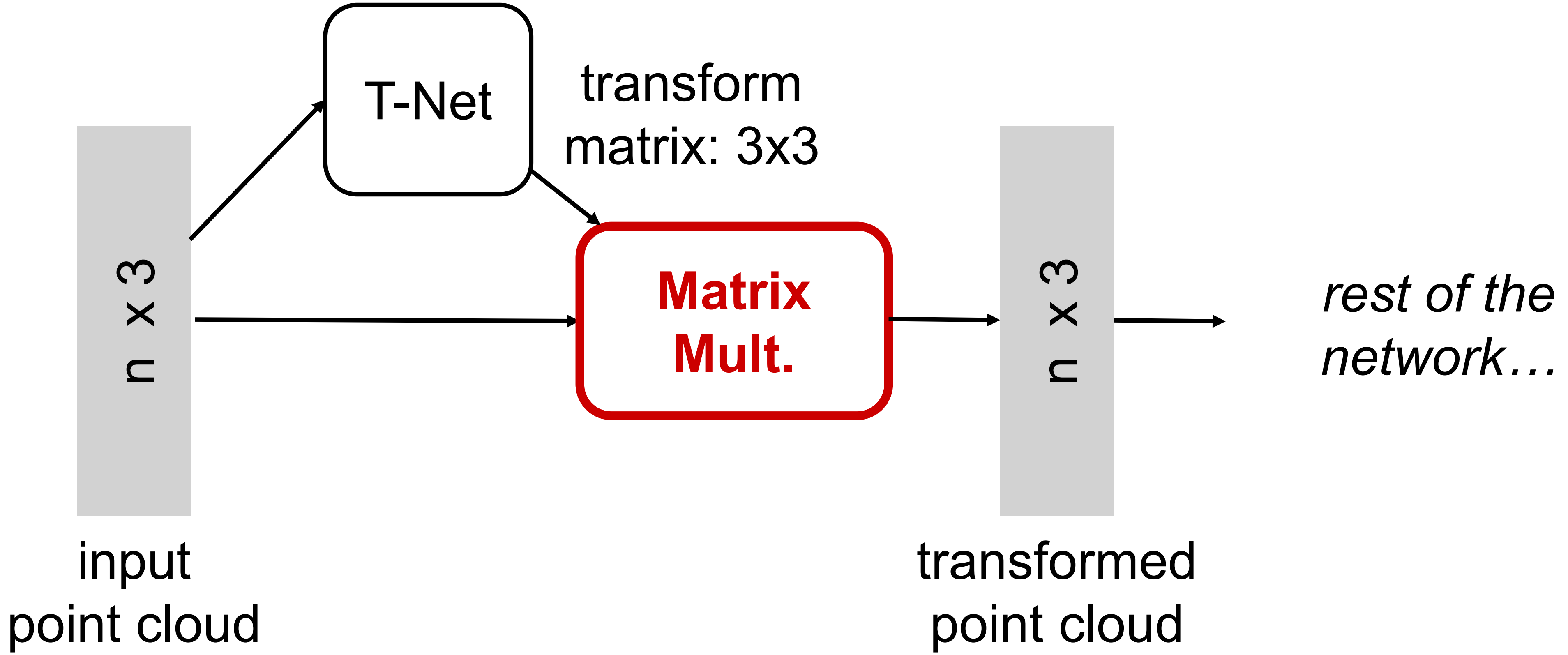
Idea: Data dependent transformation for automatic alignment



Input Alignment by Transformer Network

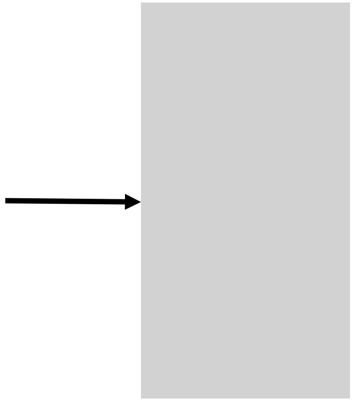
Idea: Data dependent transformation for automatic alignment

The transformation is just matrix multiplication!



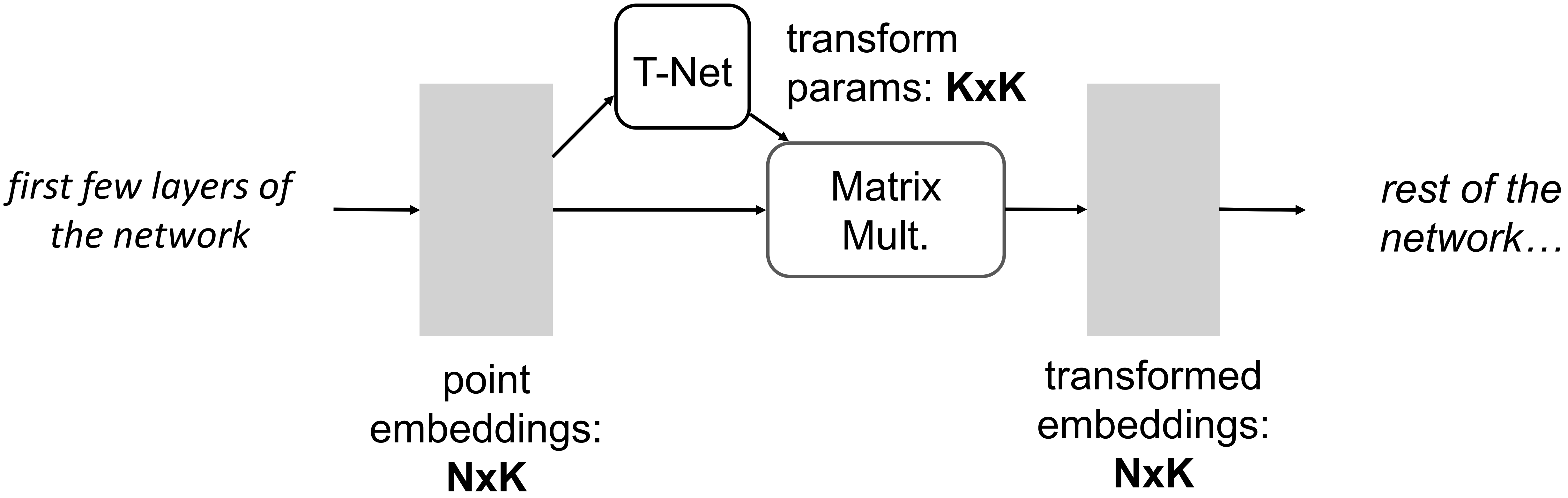
Embedding Space Alignment

first few layers of the network

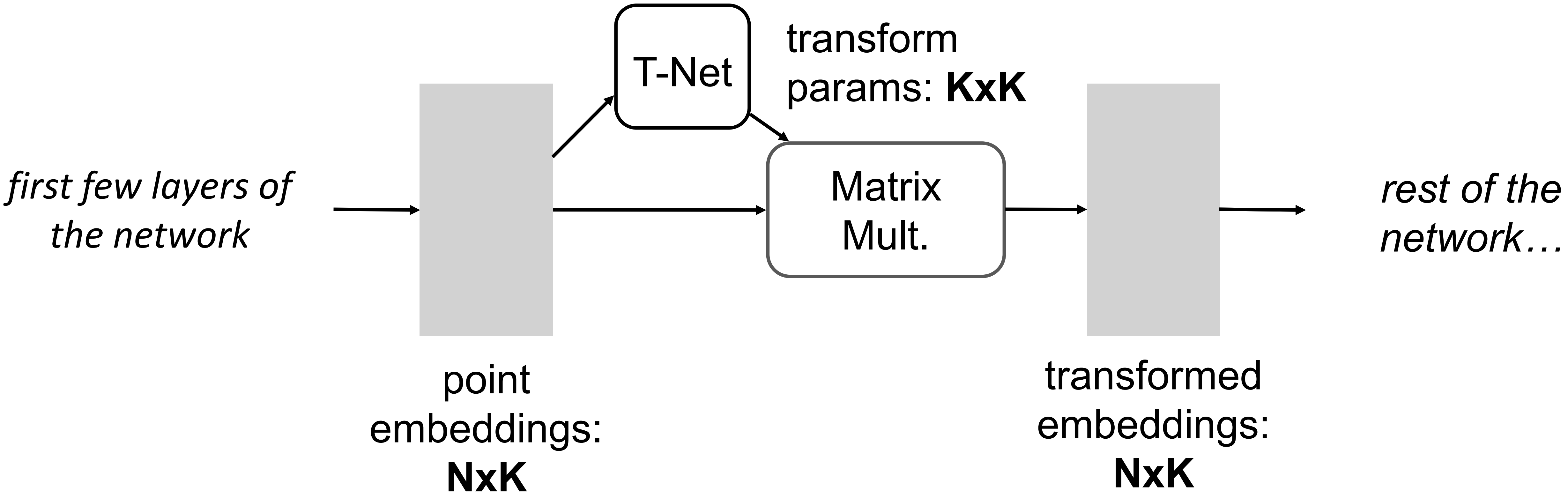


point embeddings: **NxK**

Embedding Space Alignment



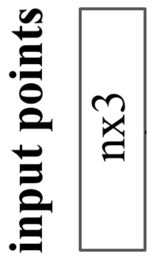
Embedding Space Alignment



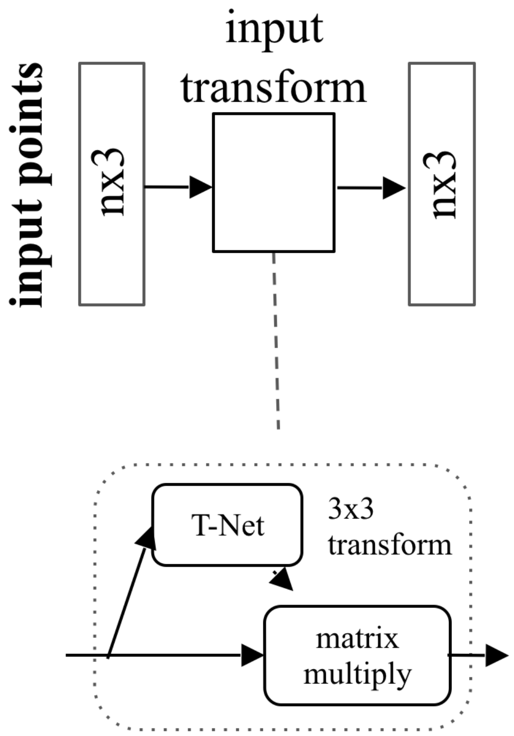
Regularization loss:

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

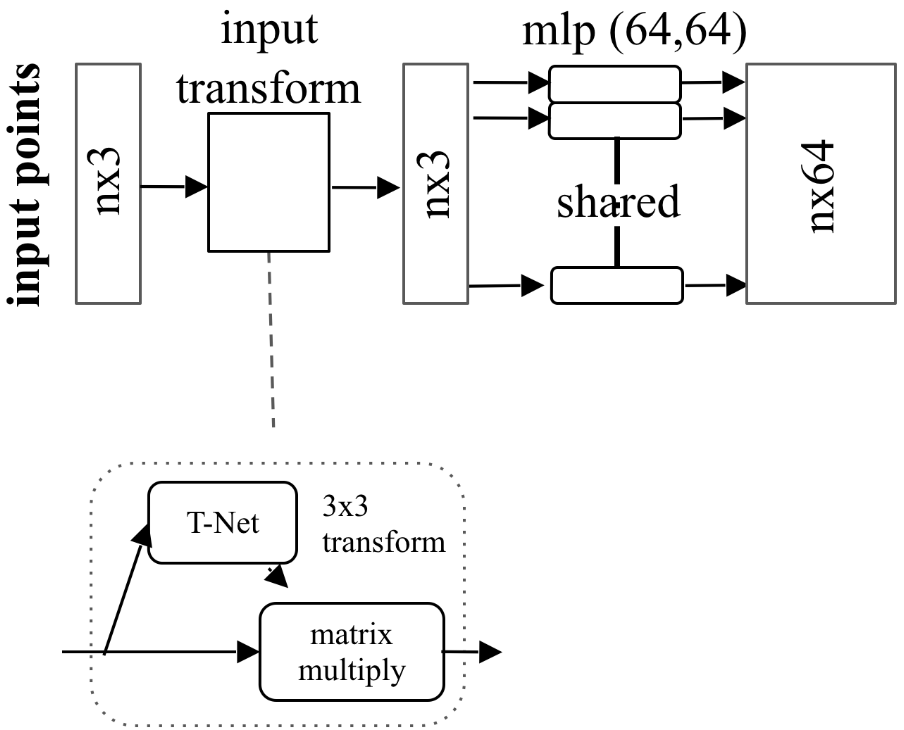
PointNet Classification Network



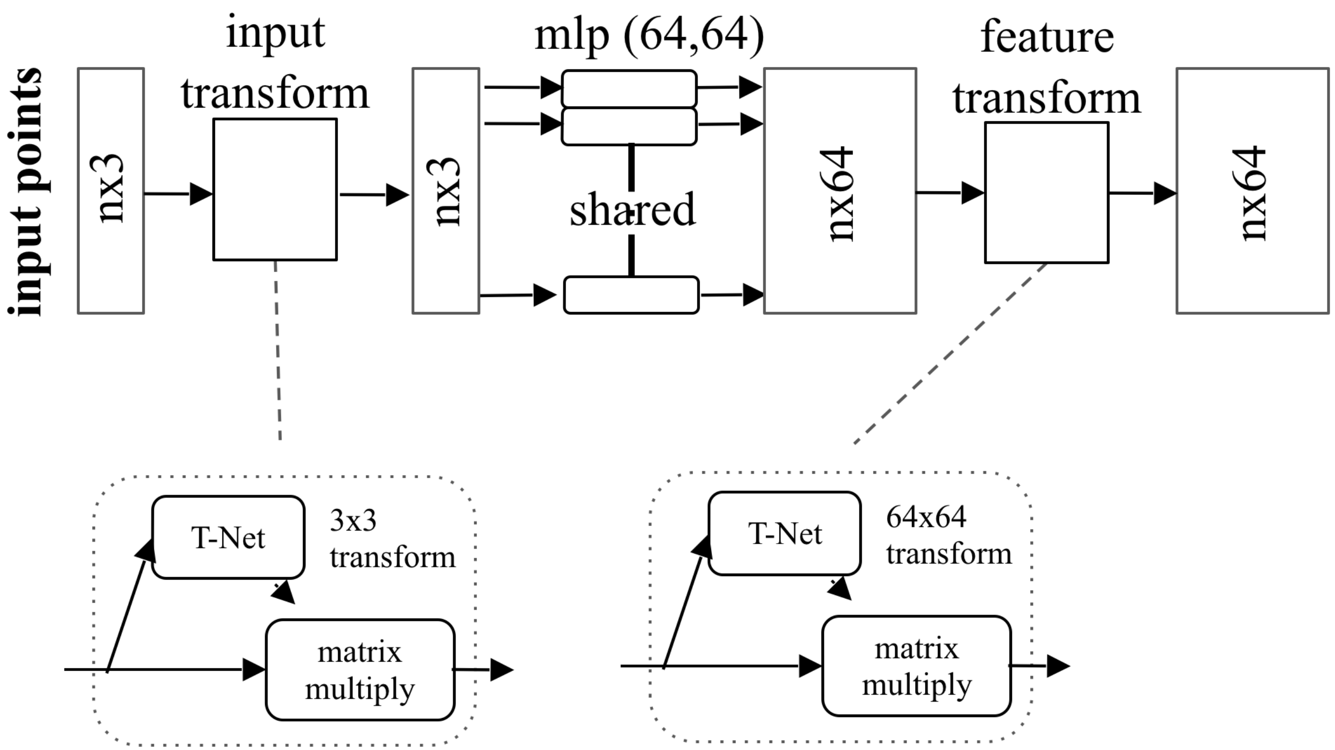
PointNet Classification Network



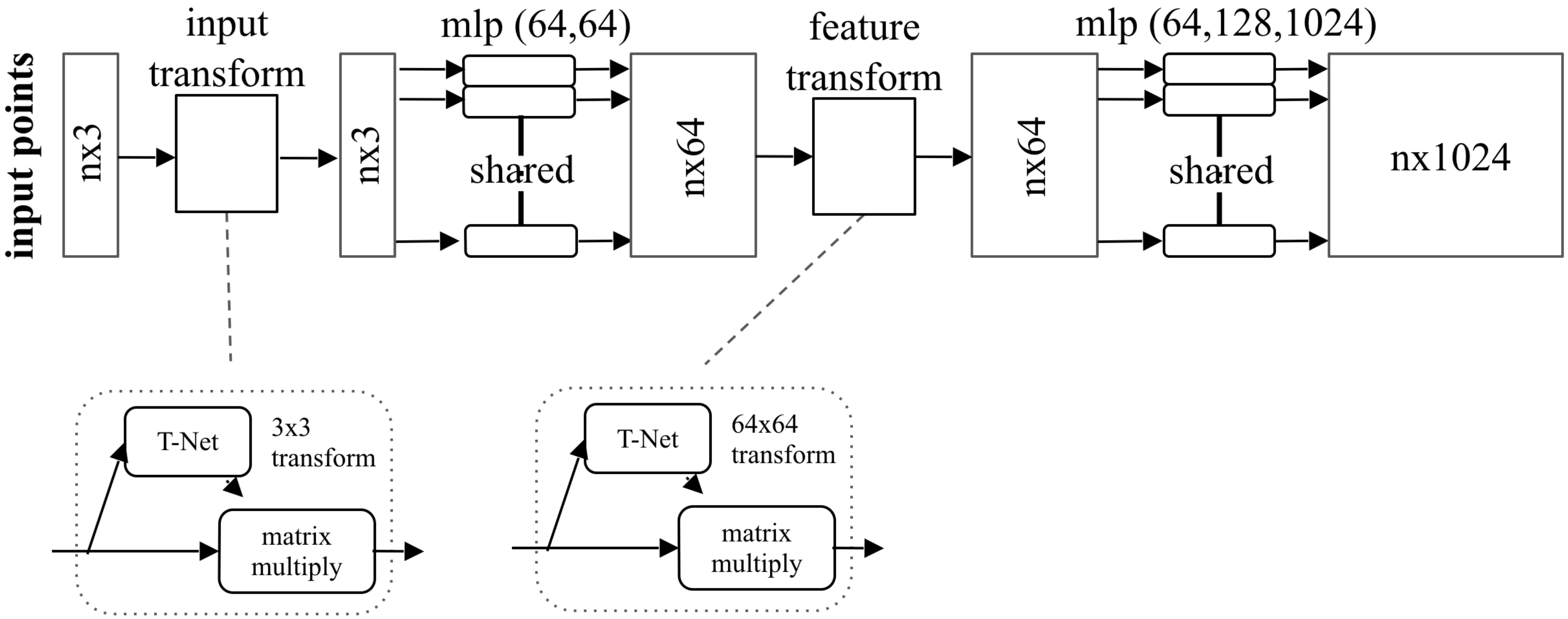
PointNet Classification Network



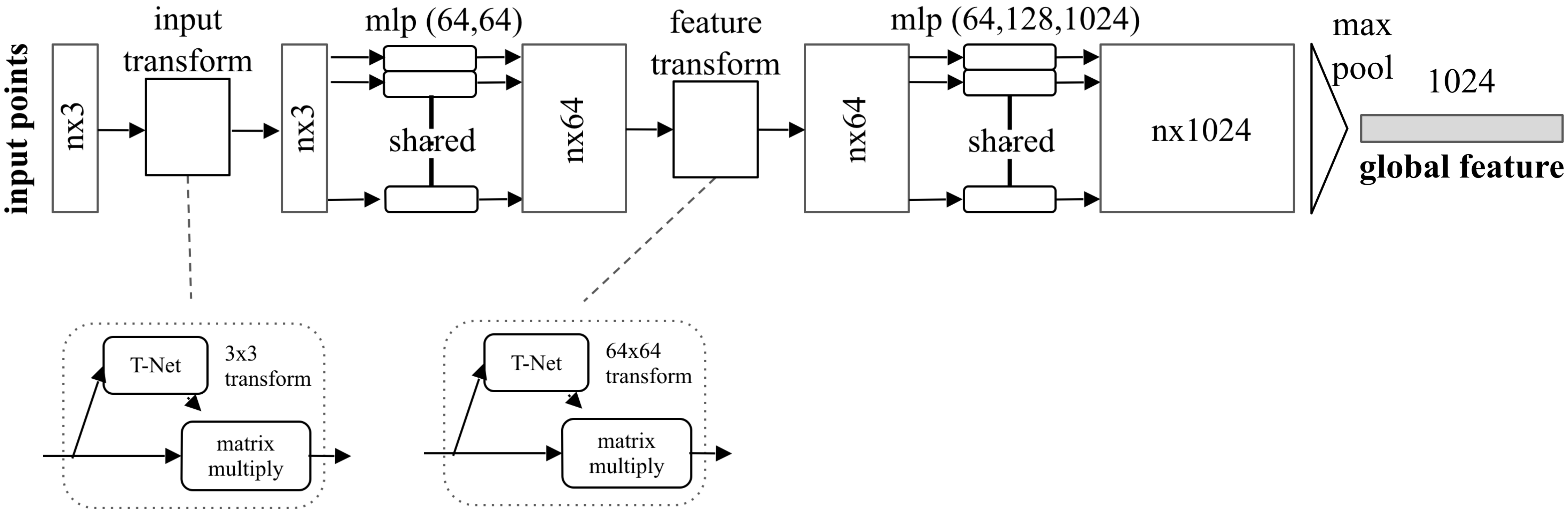
PointNet Classification Network



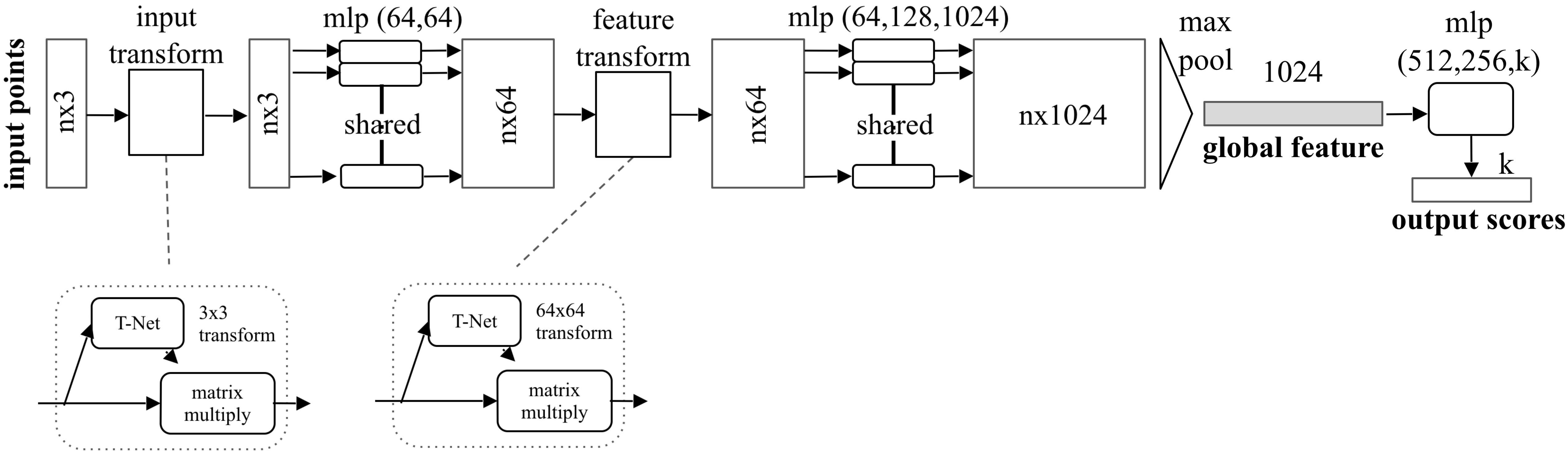
PointNet Classification Network



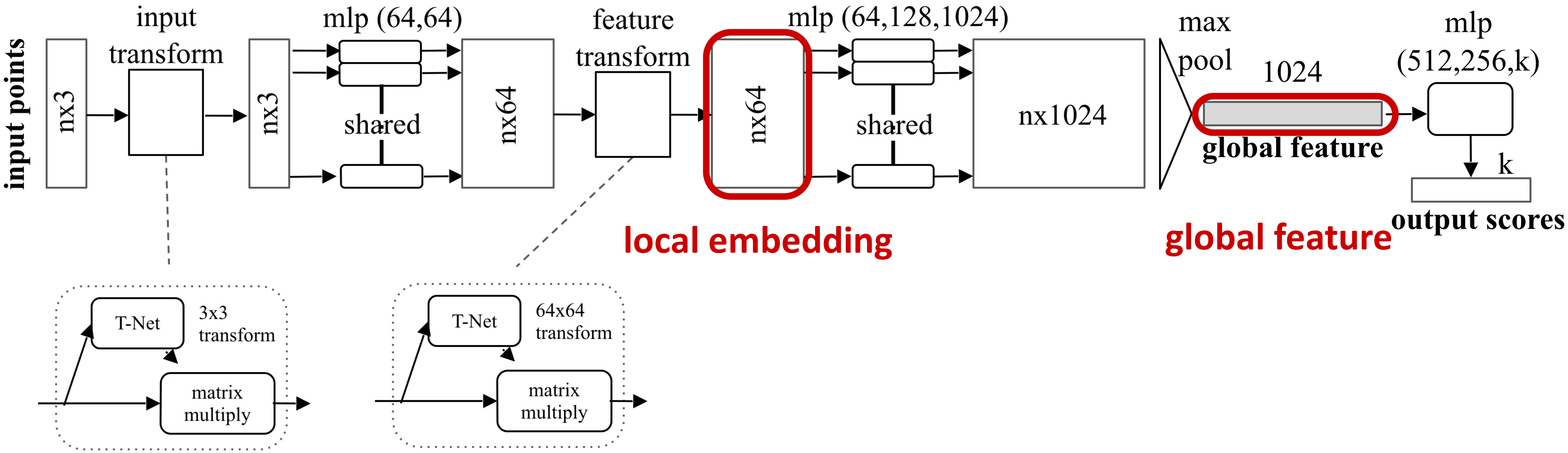
PointNet Classification Network



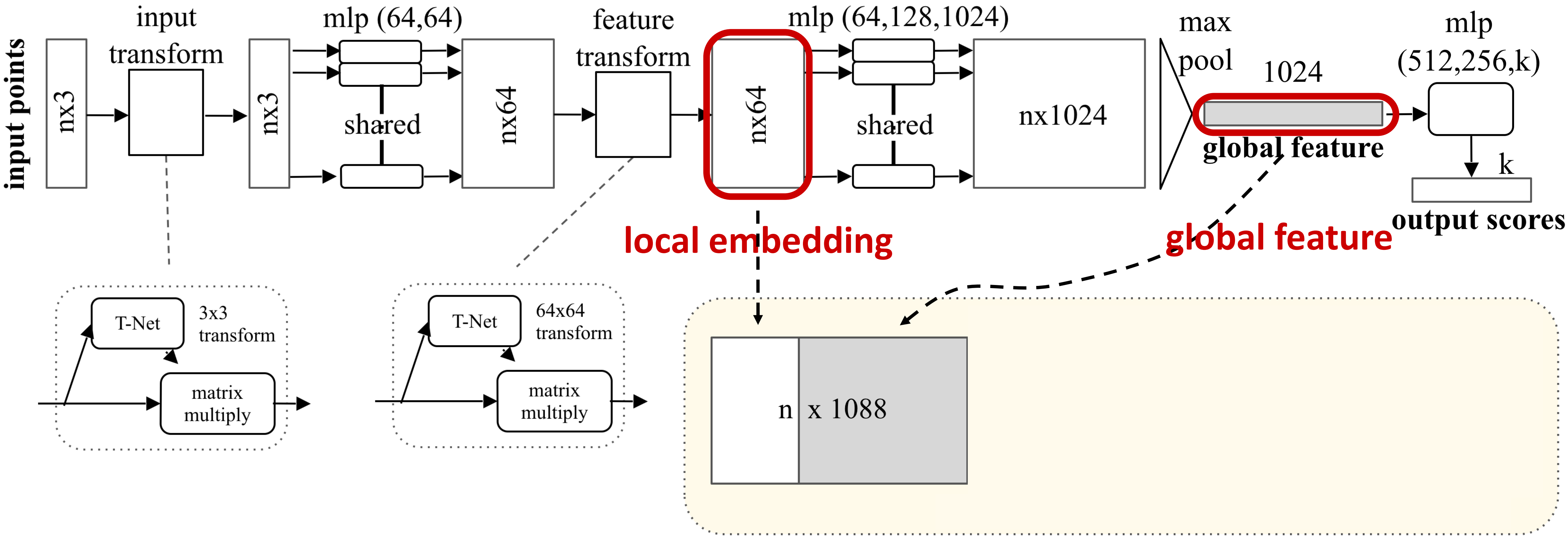
PointNet Classification Network



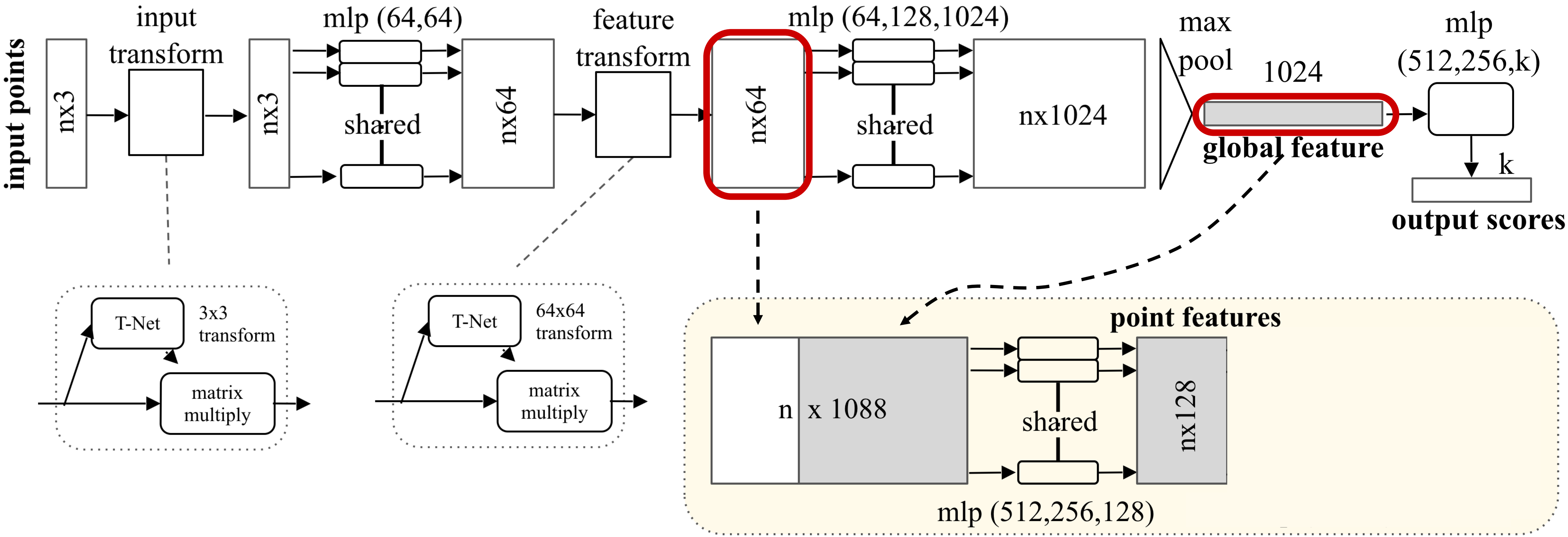
Extension to PointNet Segmentation Network



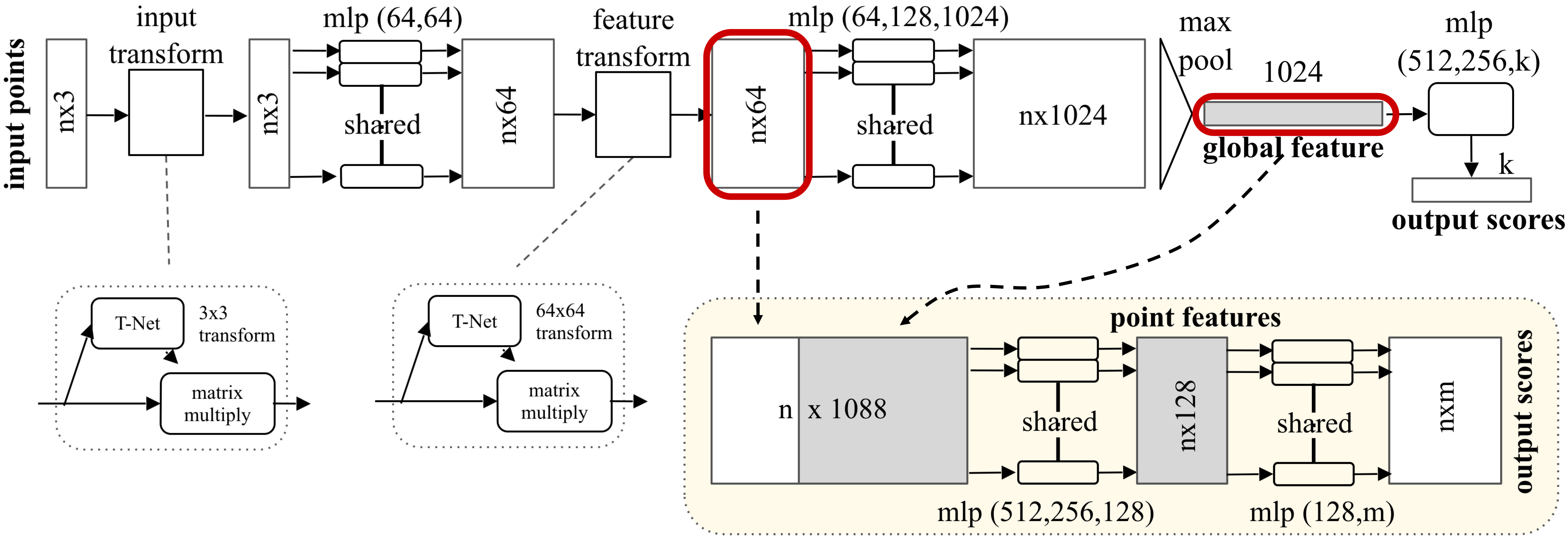
Extension to PointNet Segmentation Network



Extension to PointNet Segmentation Network



Extension to PointNet Segmentation Network



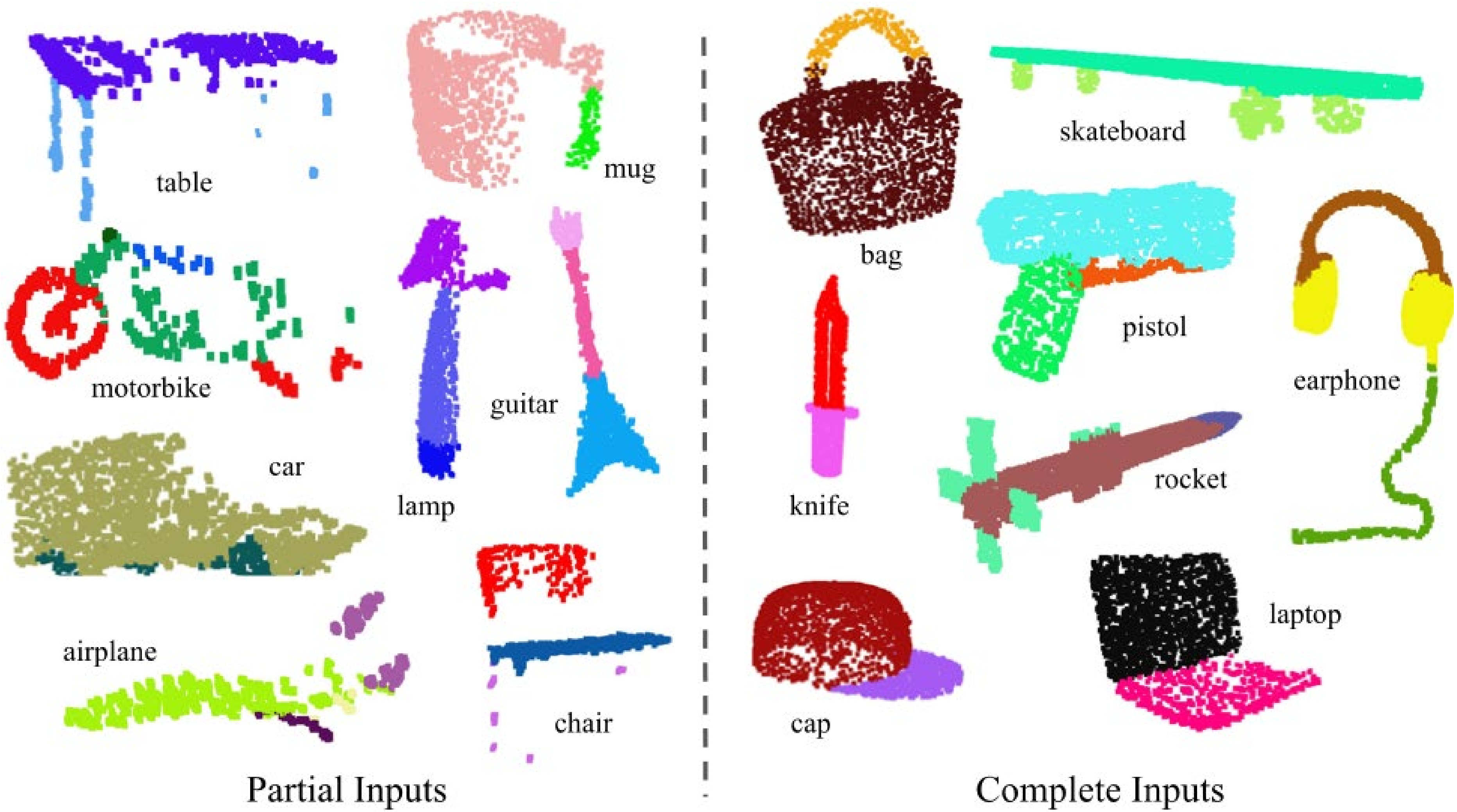
Results on Object Classification

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 (%)

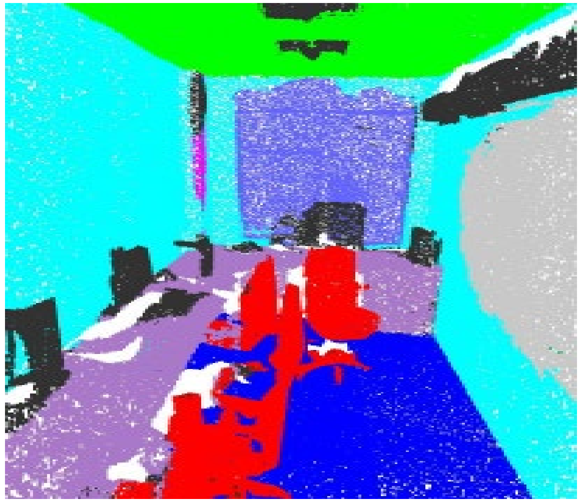
Results on Object Part Segmentation

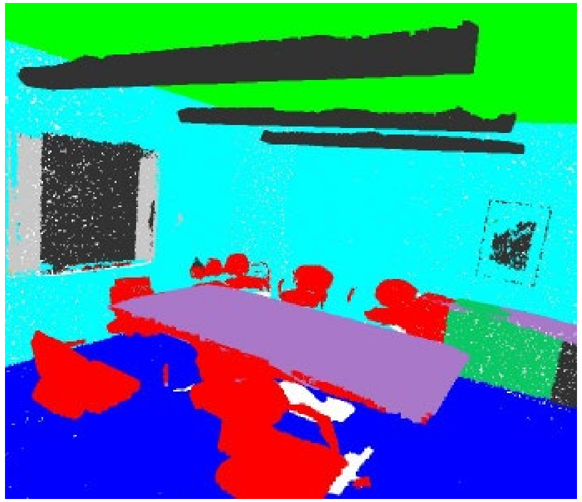


Results on Semantic Scene Parsing

Input

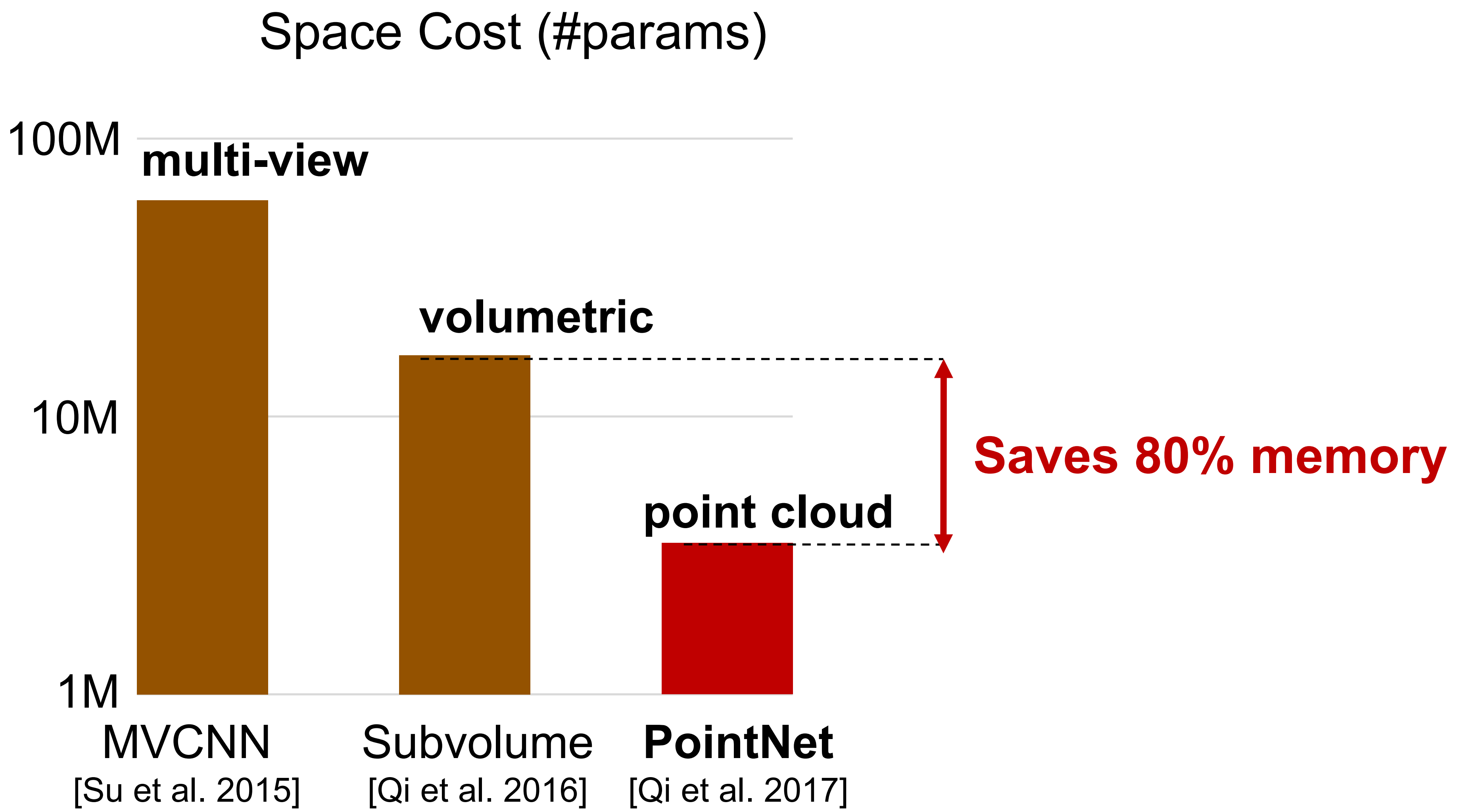
Output



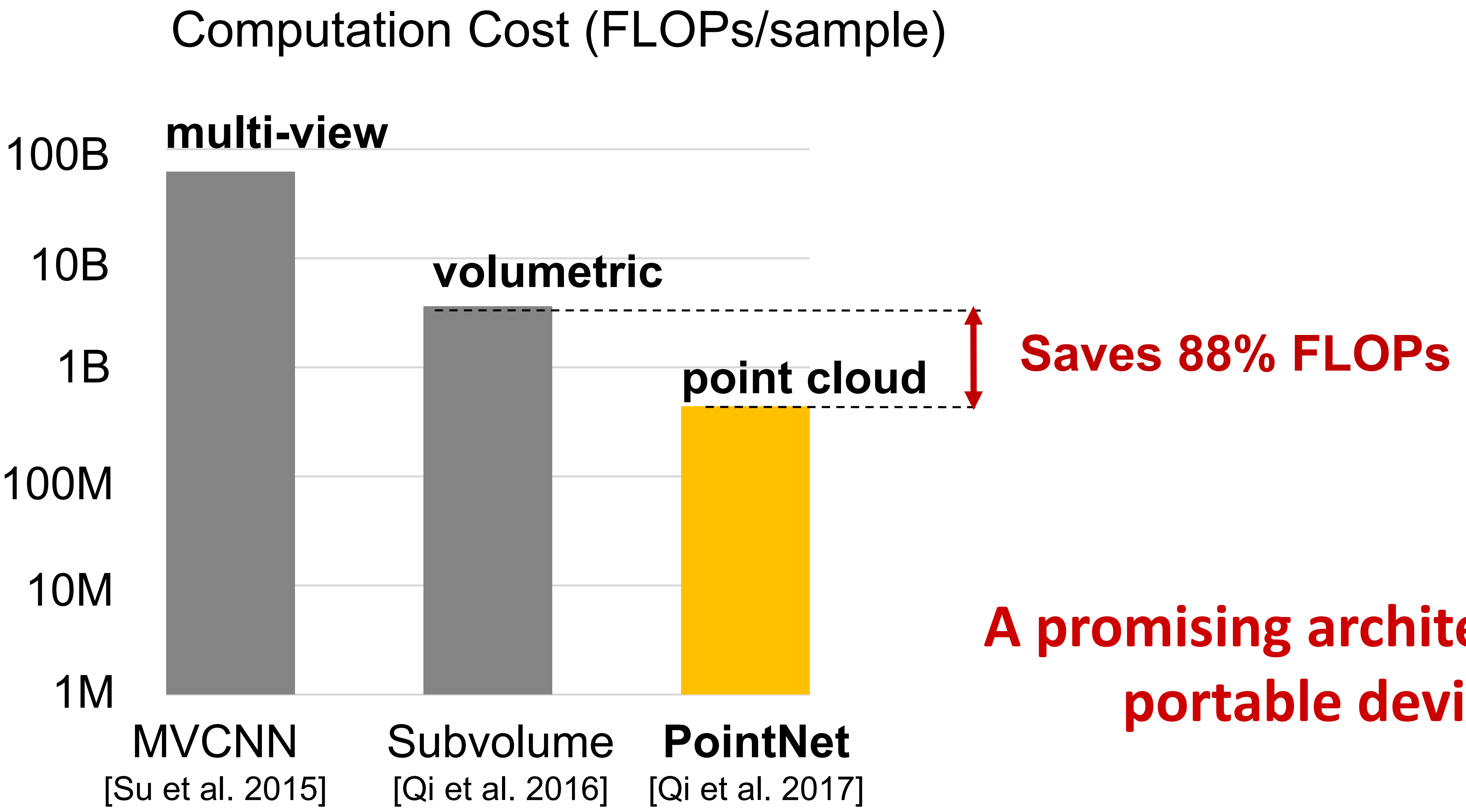


dataset: Stanford 2D-3D-S (Matterport scans)

PointNet is Light-Weight and Fast

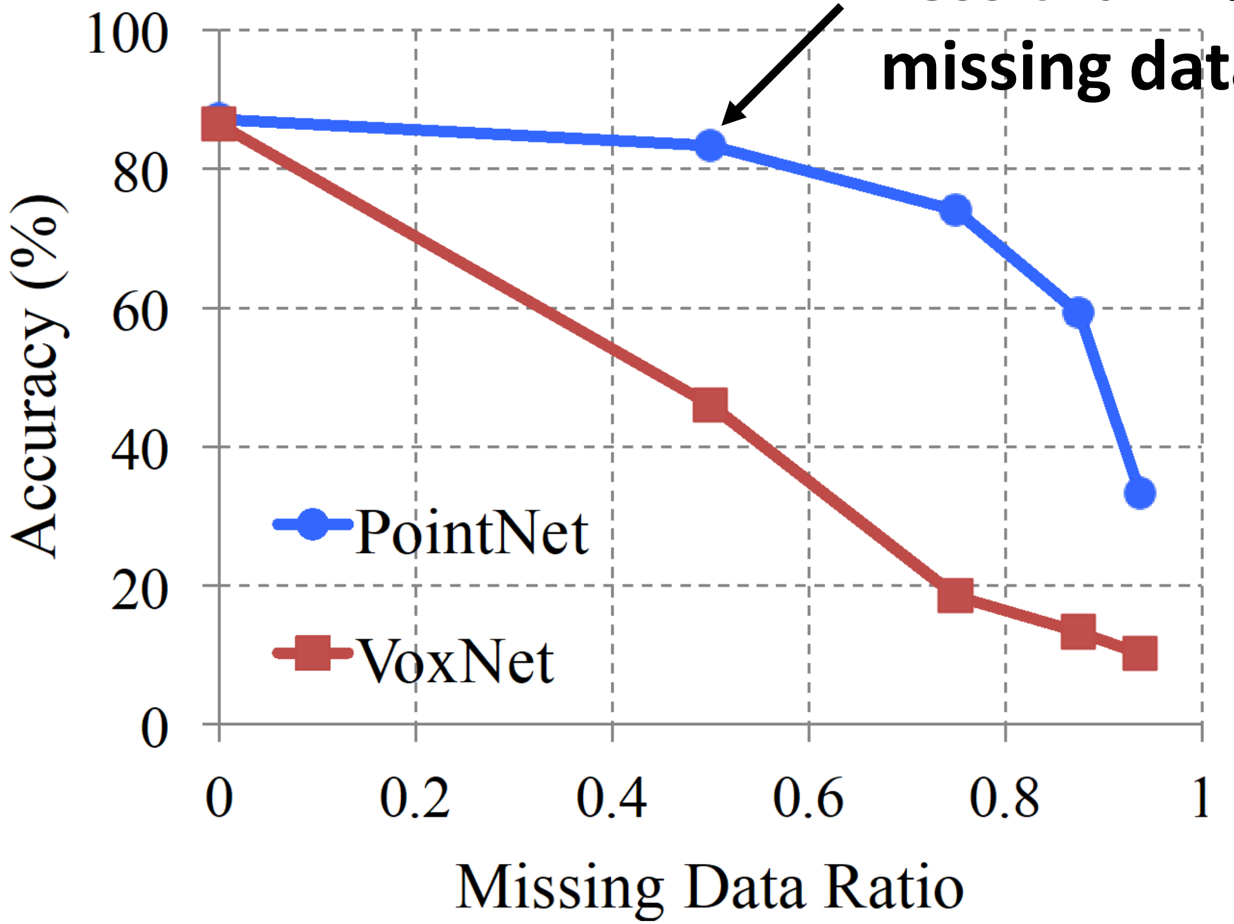


PointNet is Light-Weight and Fast



A promising architecture for portable devices!

PointNet is Robust to Data Corruption



Less than 2% accuracy drop with 50% missing data

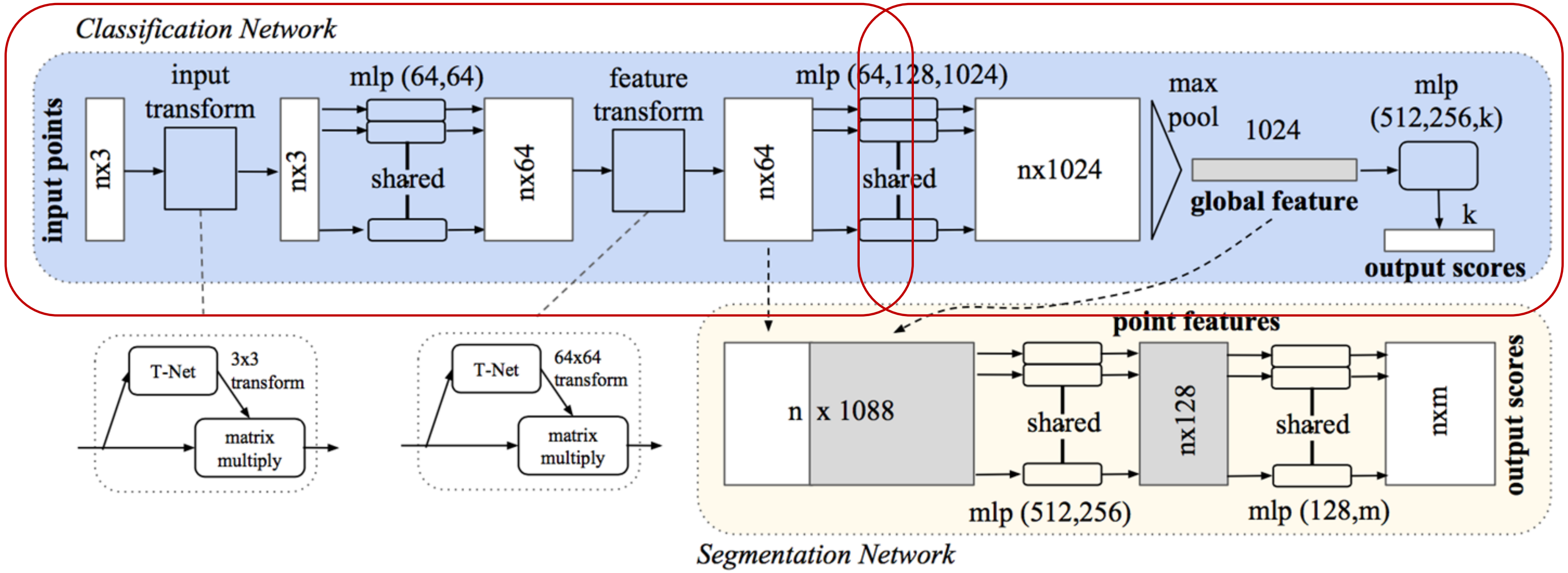
Why is PointNet so robust to missing data?

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

Visualizing Global Point Cloud Features

Original Shape

Learning Interesting Points

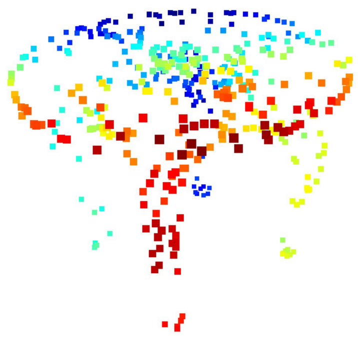


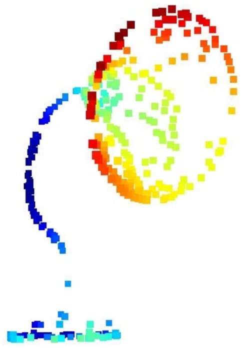
Pointnet learns optimization criteria, which in turn pick interesting points

Visualizing Global Point Cloud Features

Original Shape

Critical Points





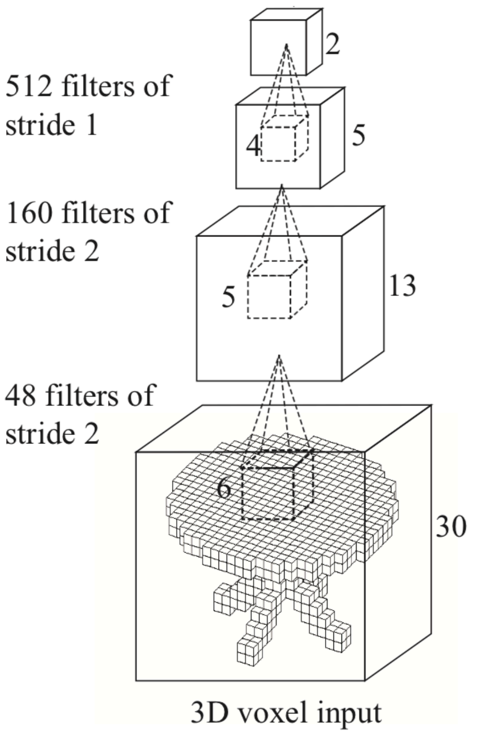
PointNet *learns to pick perceptually interesting points*

A semantic *core-set …*

From PointNet to PointNet++

Limitations of PointNet

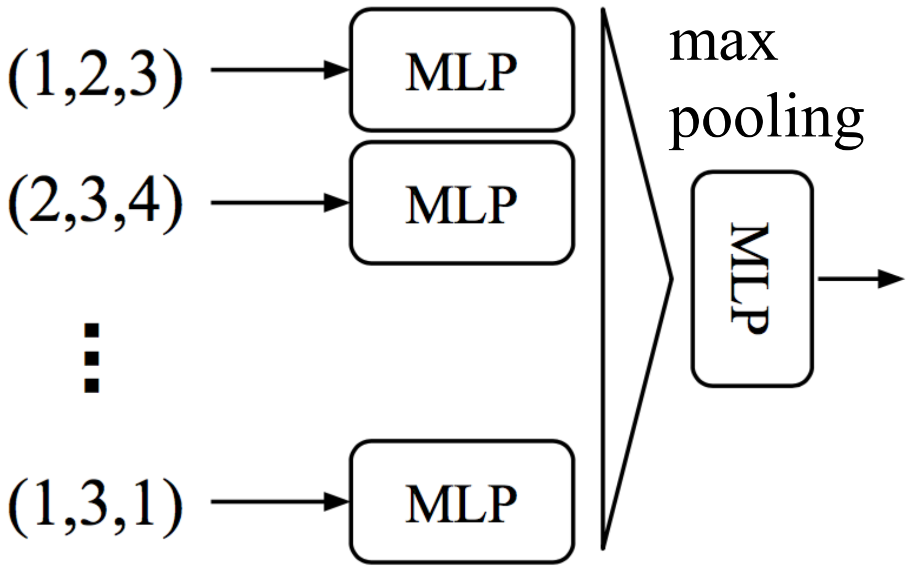
Hierarchical feature learning
multiple levels of abstraction



3D CNN [Wu et al.2015]

v.s.

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

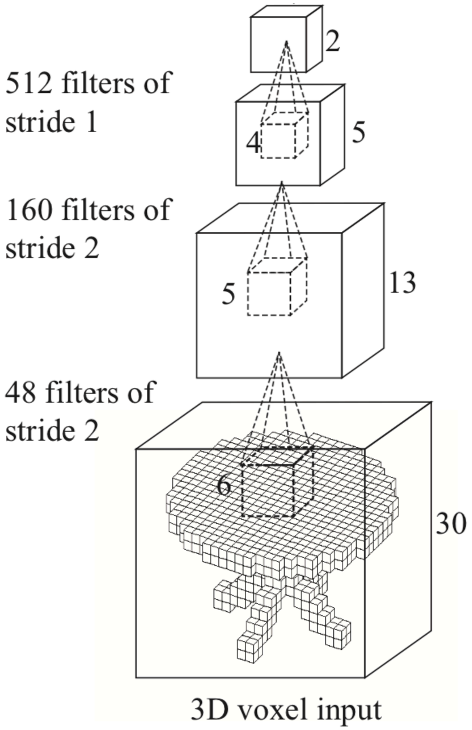


PointNet (vanilla) [Qi et al.2017]

Limitations of PointNet

Hierarchical feature learning

multiple levels of abstraction

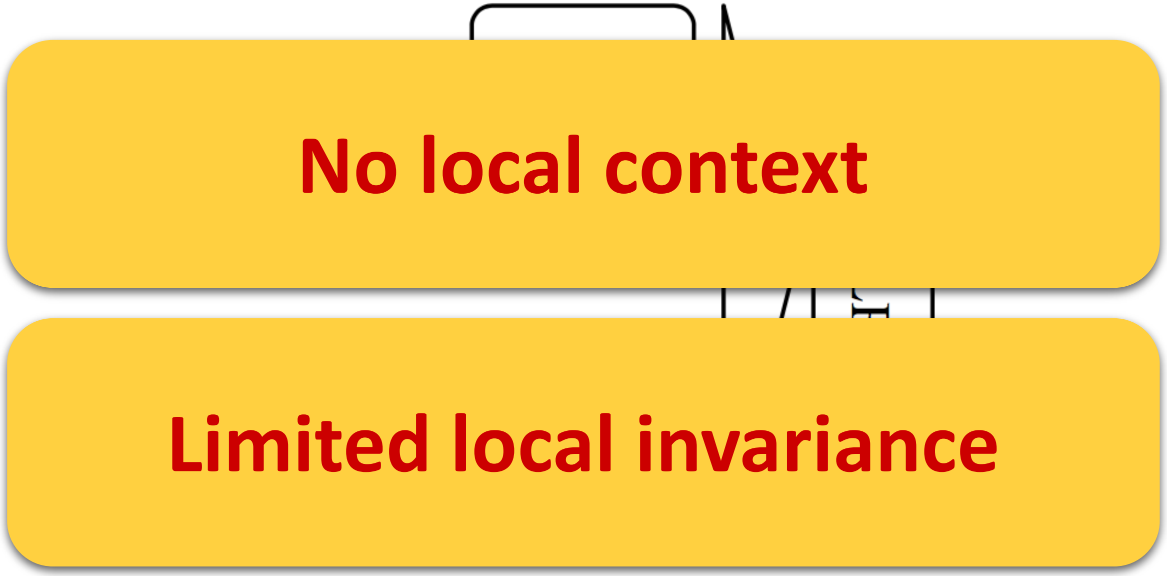


3D CNN [Wu et al.2015]

v.s.

Global feature learning

either **one** point or **all** points

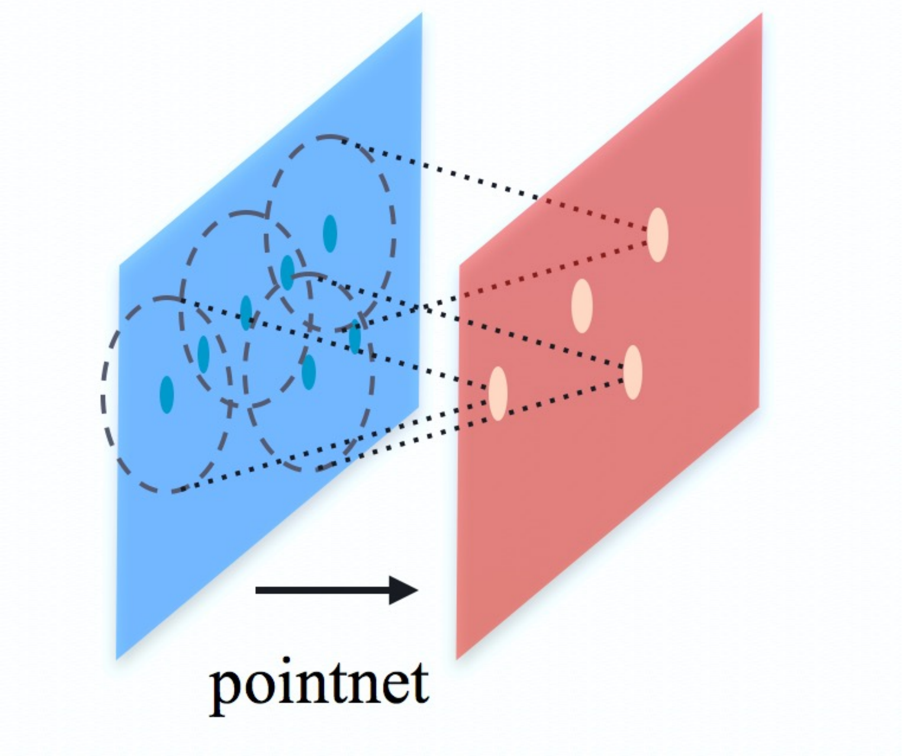


PointNet (vanilla) [Qi et al.2017]

PointNet++

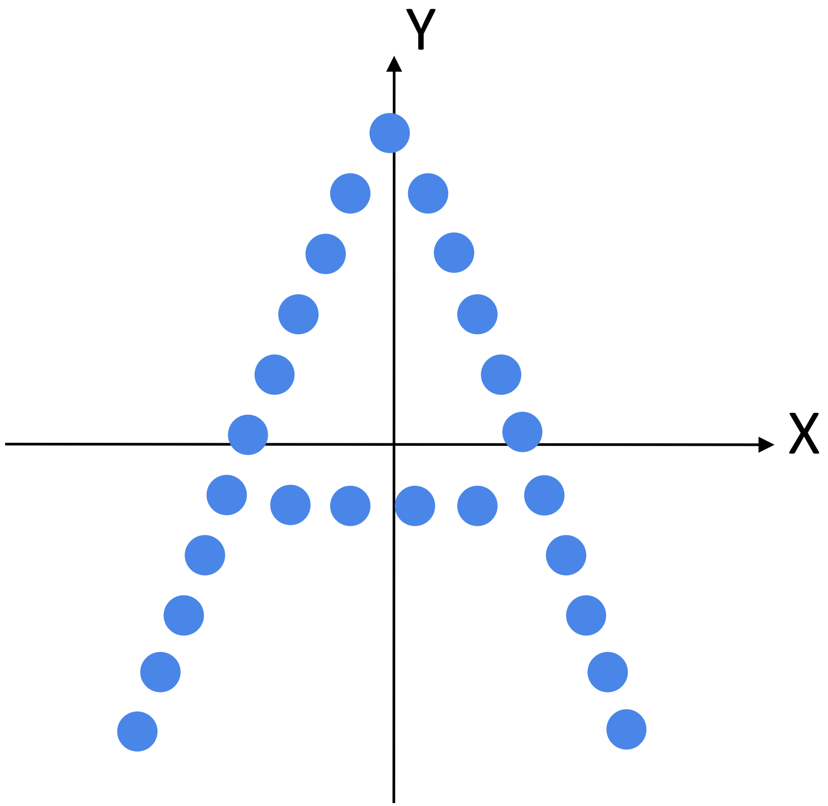
Basic idea: Recursively apply pointnet at local regions.

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



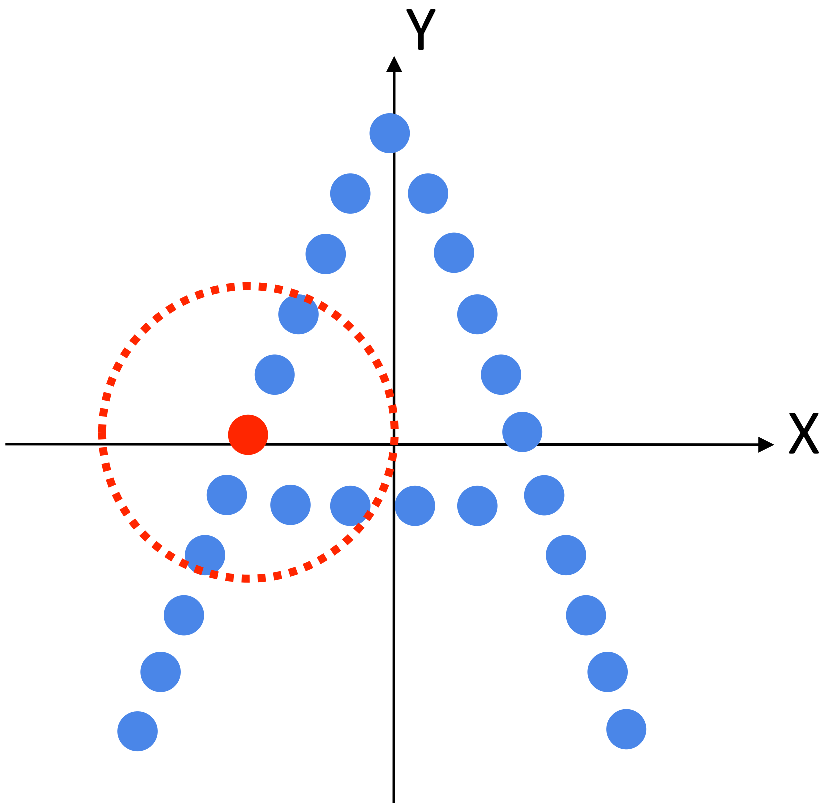
Charles R. Qi, Li Yi, Hao Su, Leonidas Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space (NIPS’17)

Hierarchical Point Feature Learning



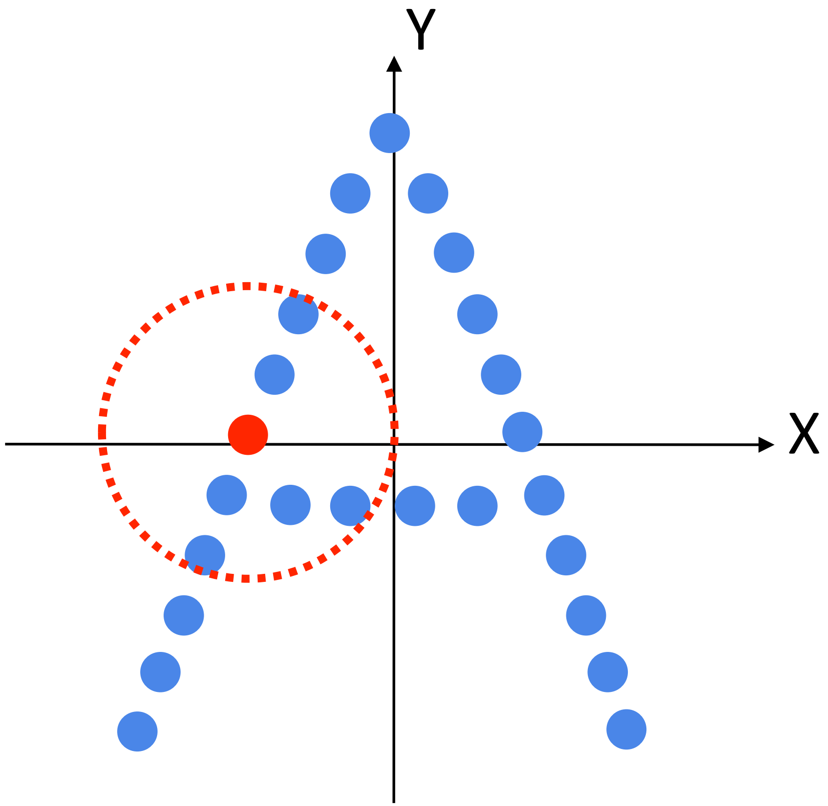
N points in (X,Y)

Hierarchical Point Feature Learning

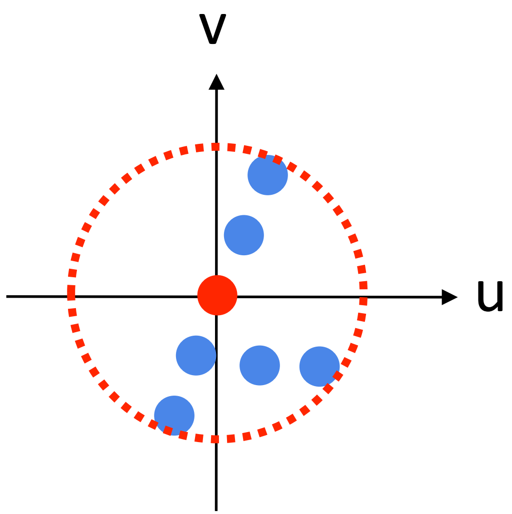


N points in (X,Y)

Hierarchical Point Feature Learning

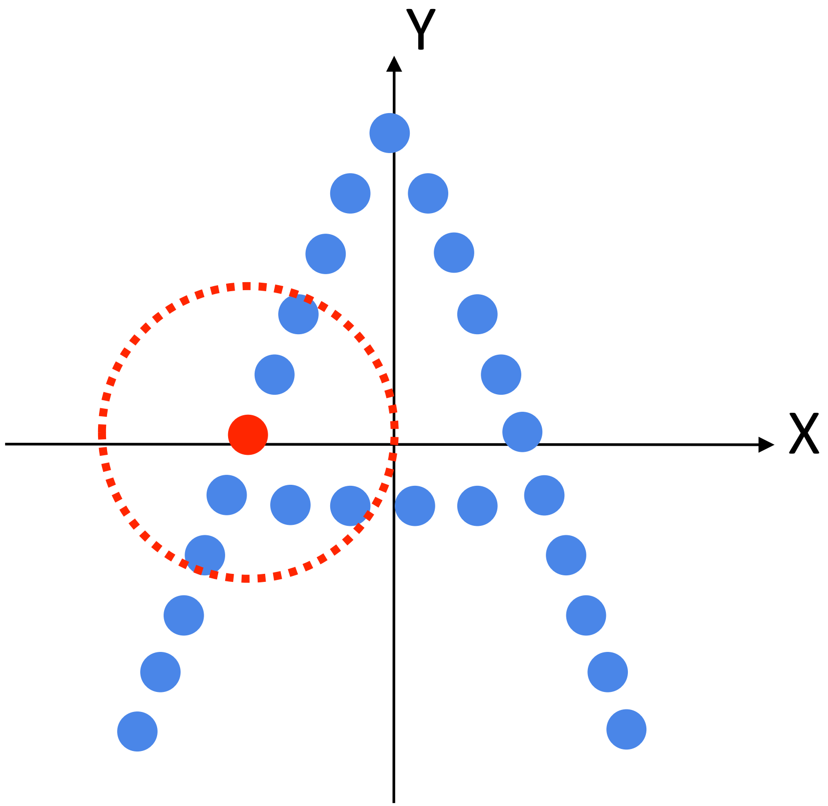


N points in (X,Y)



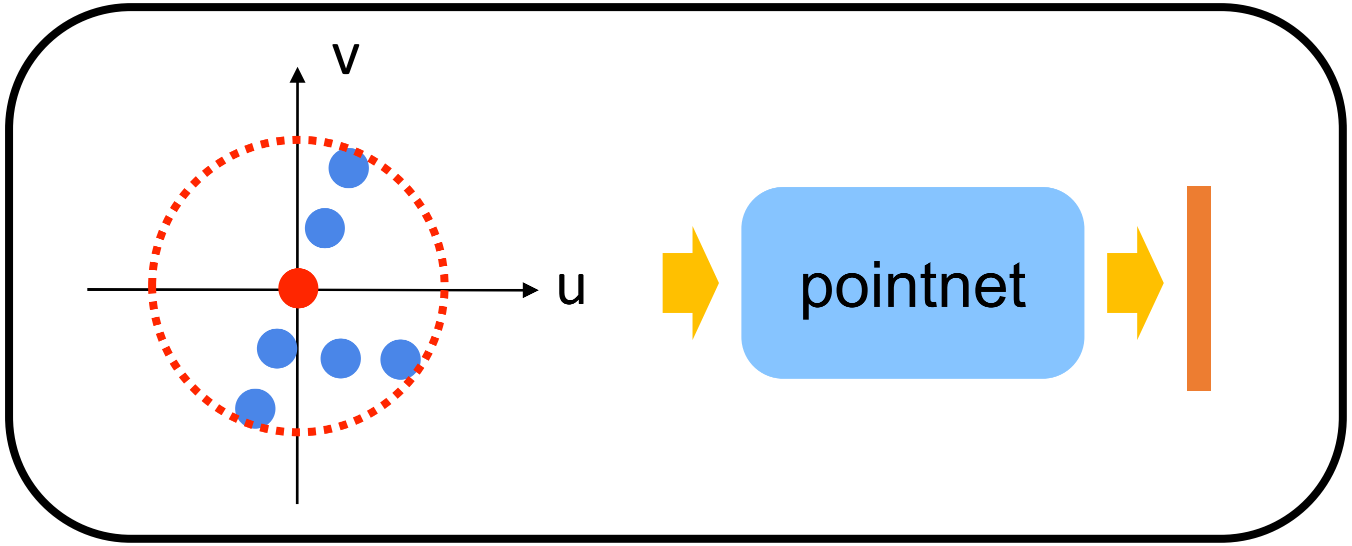
k points in local coordinates (u,v)

Hierarchical Point Feature Learning



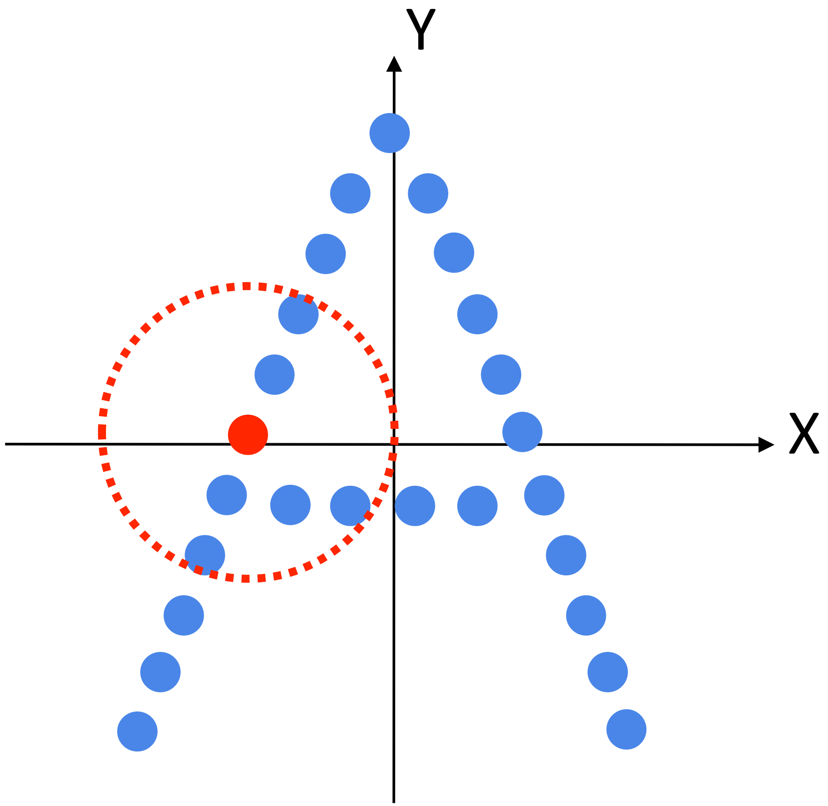
N points in (X,Y)

Apply pointnet at a local region

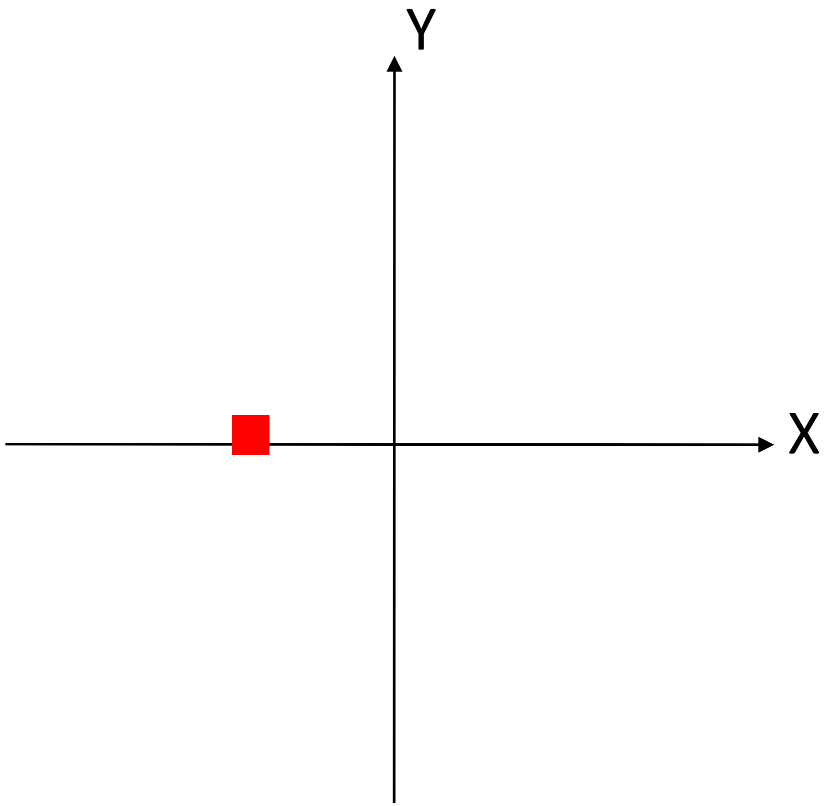


k points in local coordinates (u,v)

Hierarchical Point Feature Learning



N points in (X,Y)

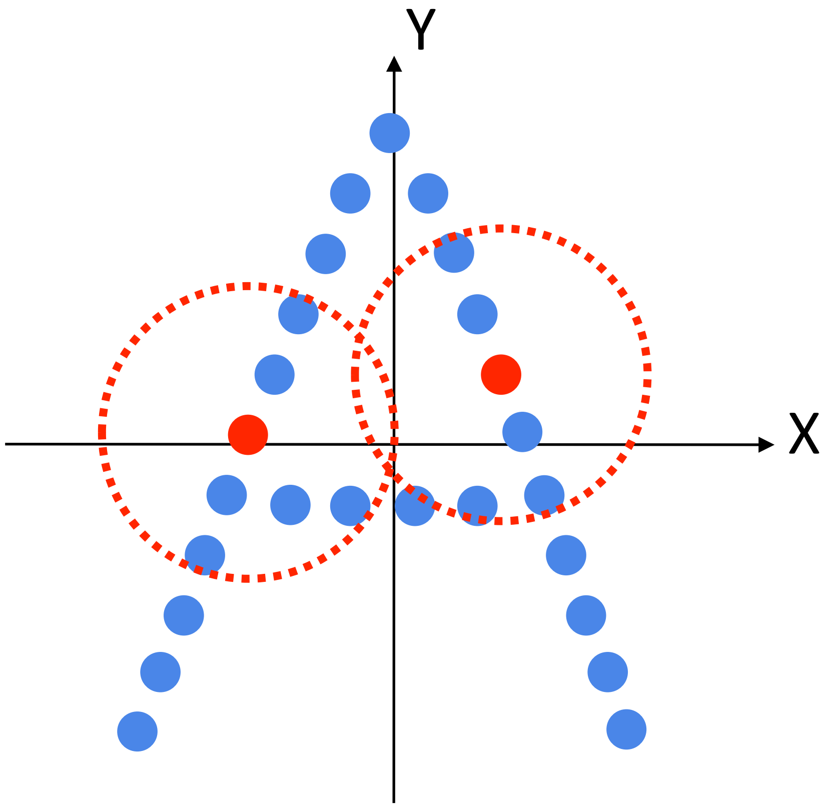


points in (X,Y, **F**)

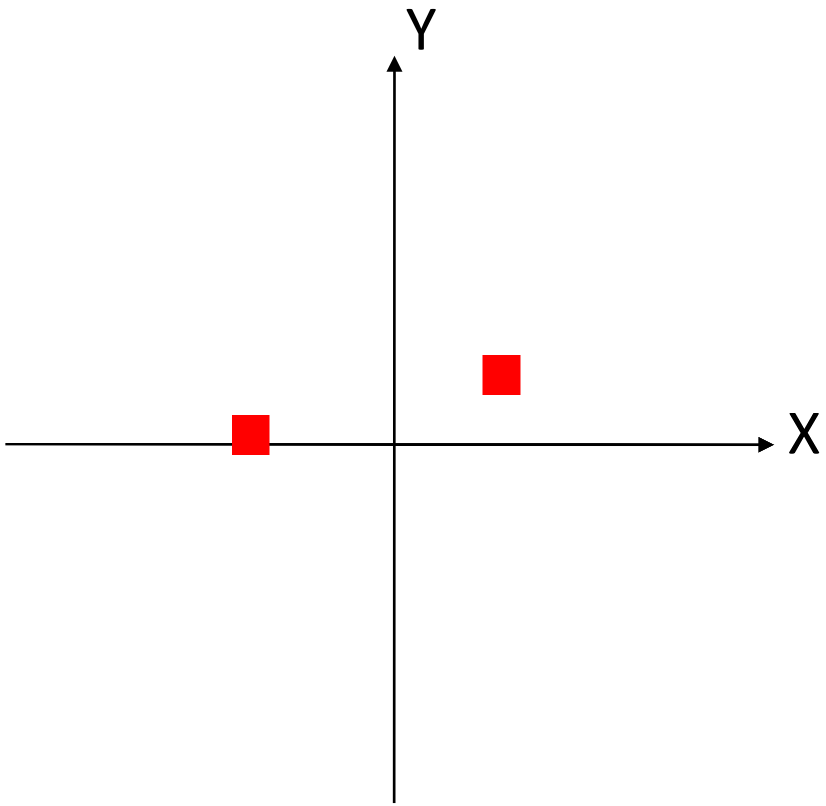
Euclidean space

high-dim feature space

Hierarchical Point Feature Learning

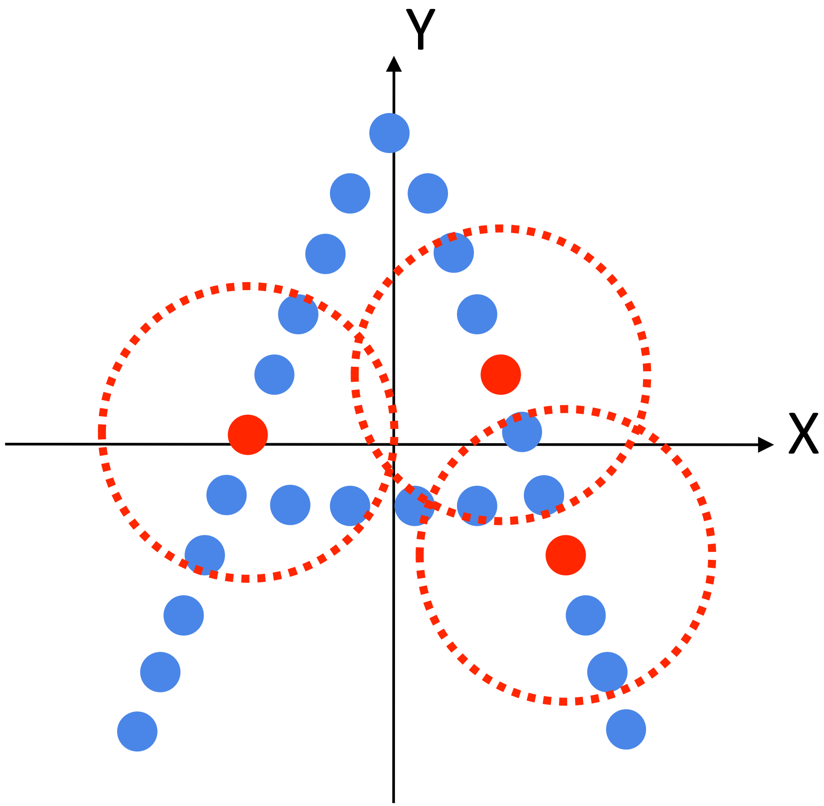


N points in (X,Y)

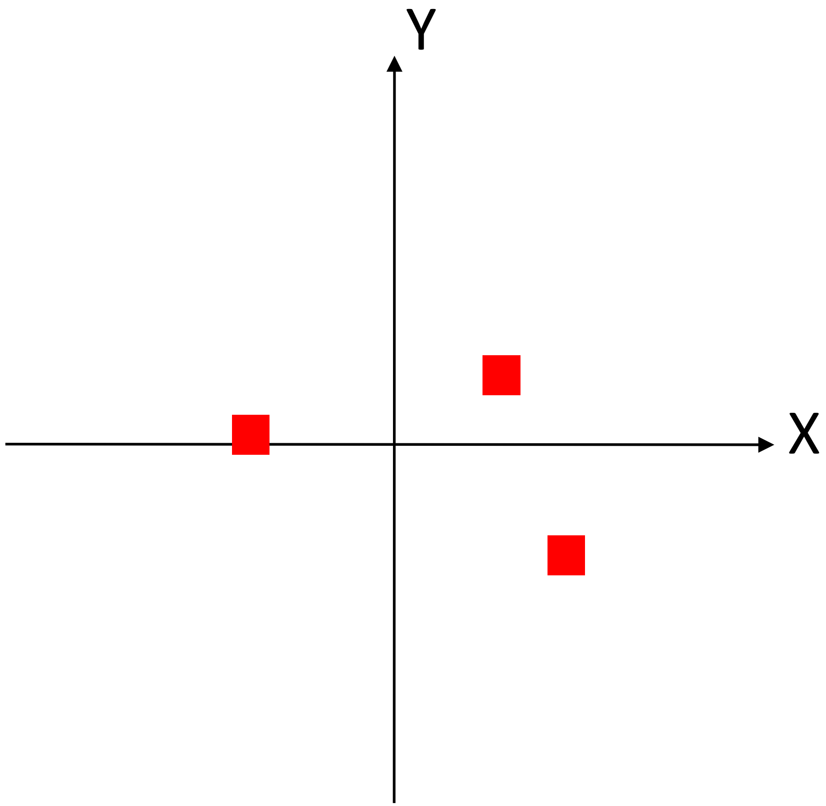


points in (X,Y, **F**)

Hierarchical Point Feature Learning

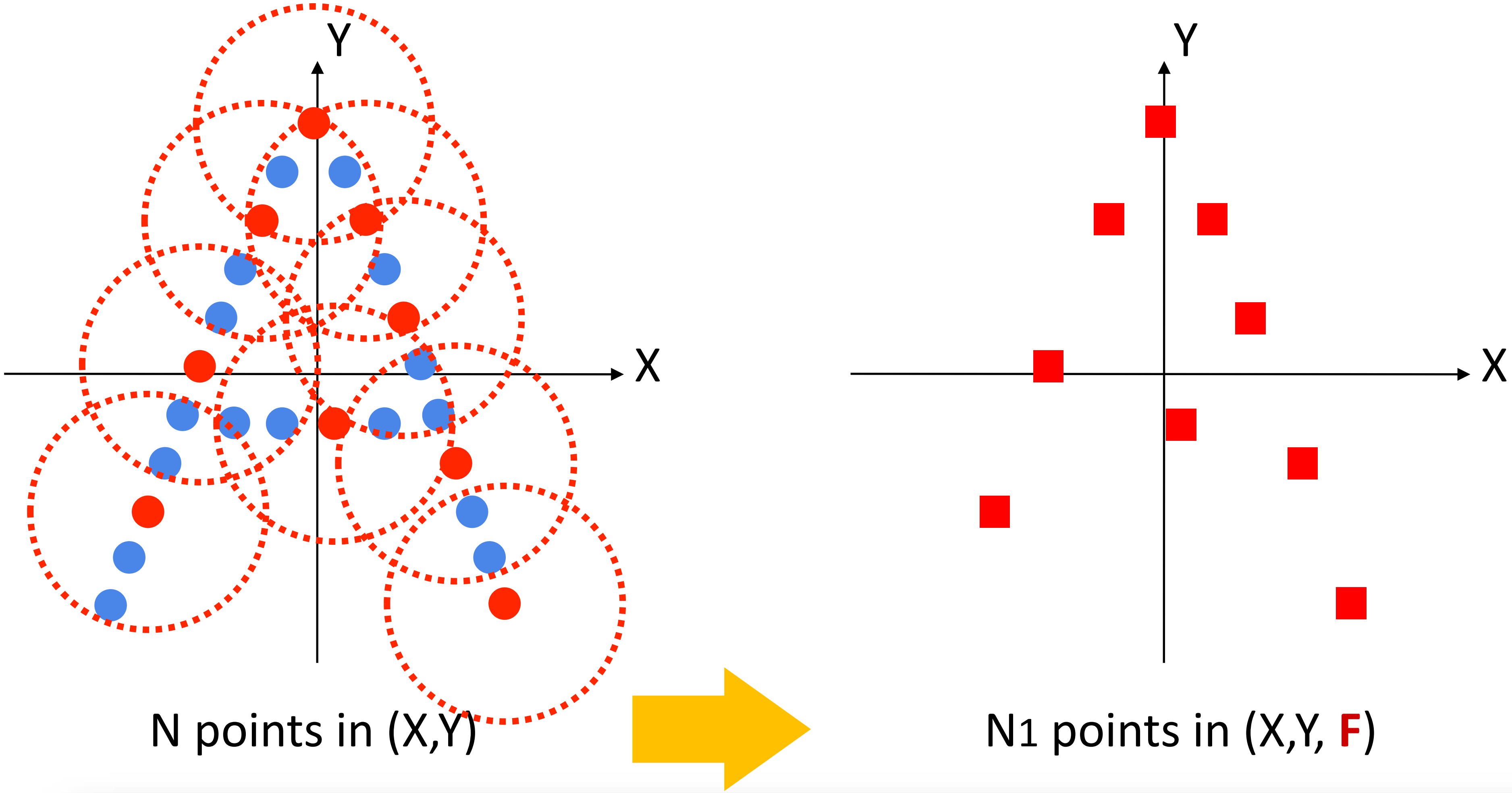


N points in (X,Y)



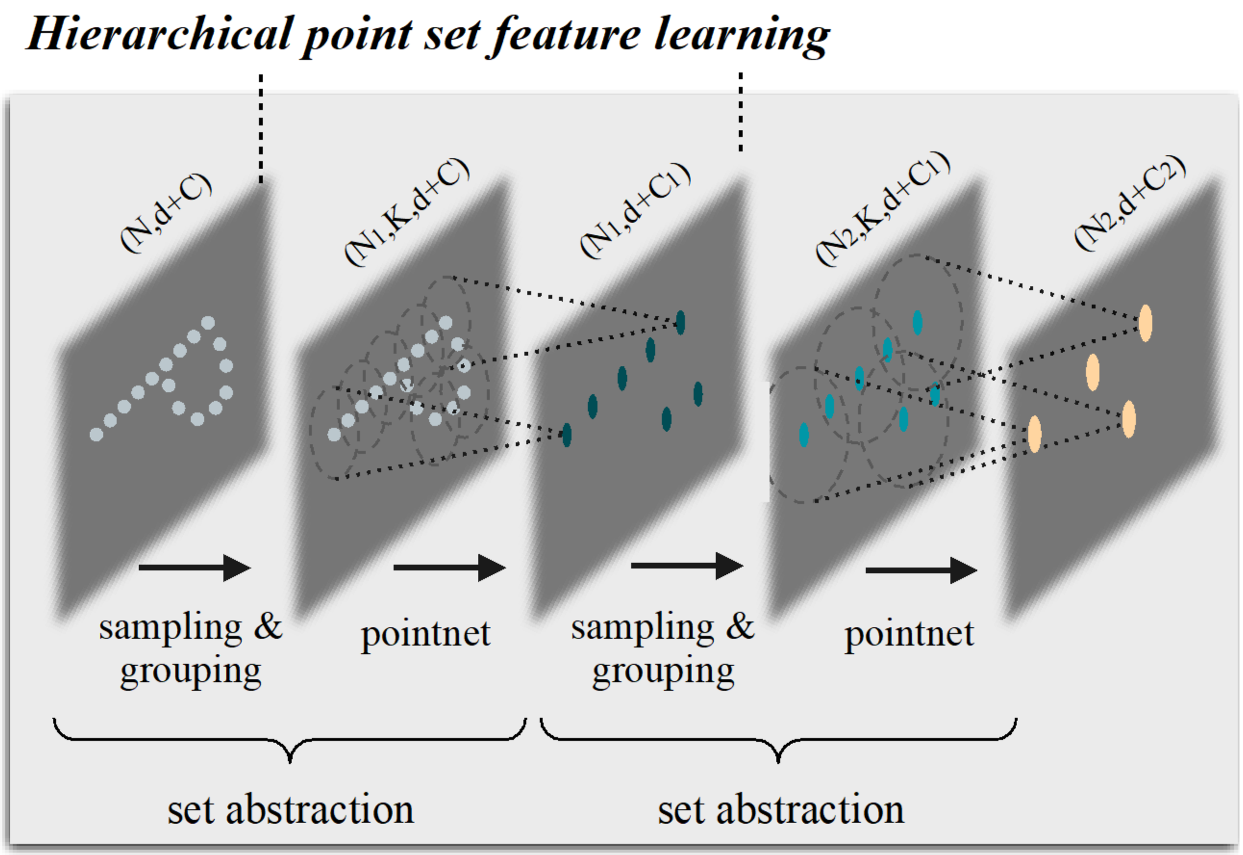
points in (X,Y, **F**)

Hierarchical Point Feature Learning



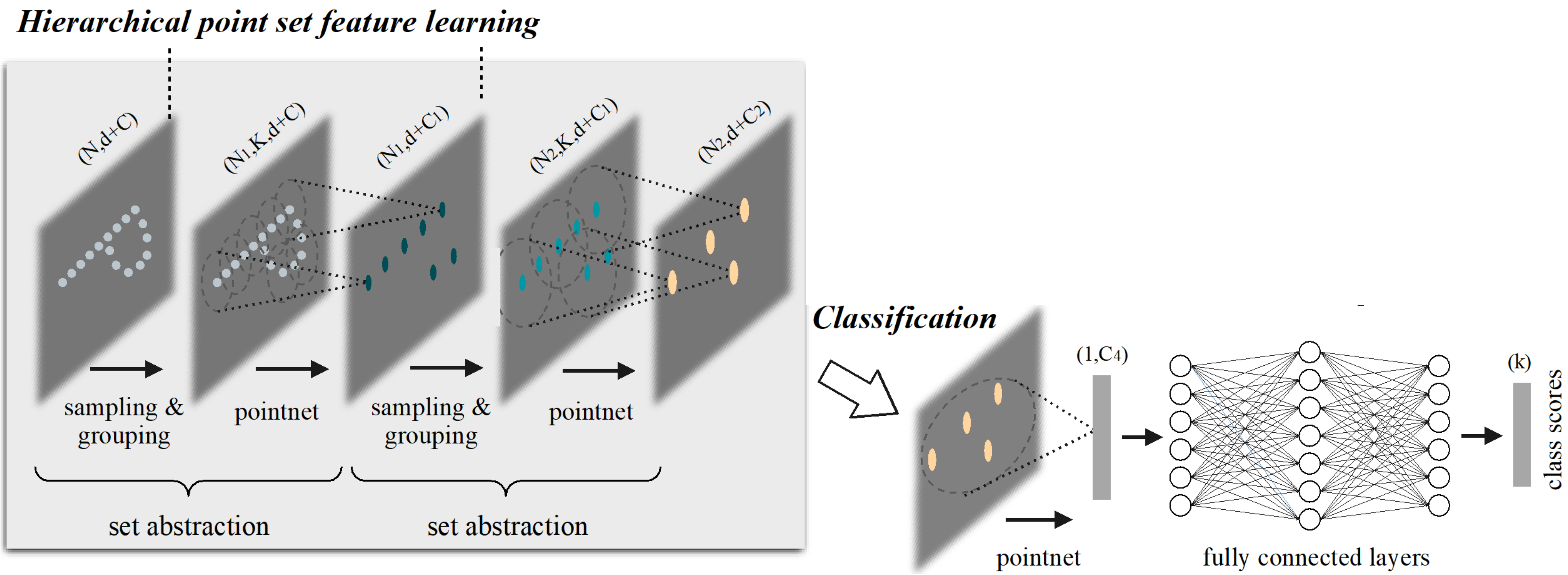
Set Abstraction: farthest point sampling + grouping + pointnet

PointNet++ for Classification and Segmentation

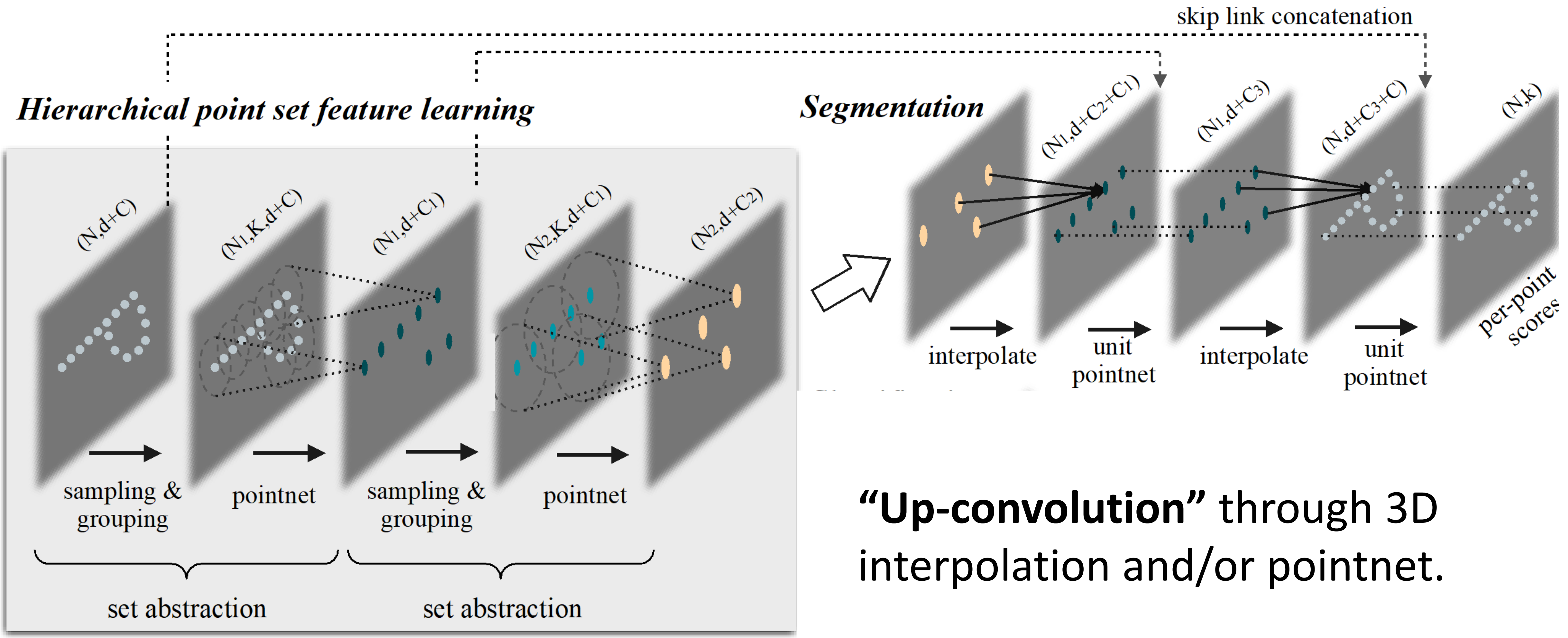


Caveat: Shouldn't feature dimensions from the lower layers affect connectivity at the higher layers?

PointNet++ for Classification and Segmentation



PointNet++ for Classification and Segmentation

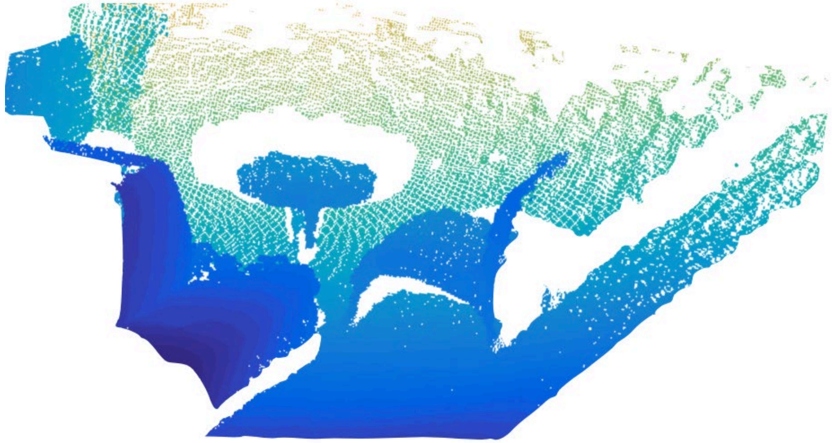


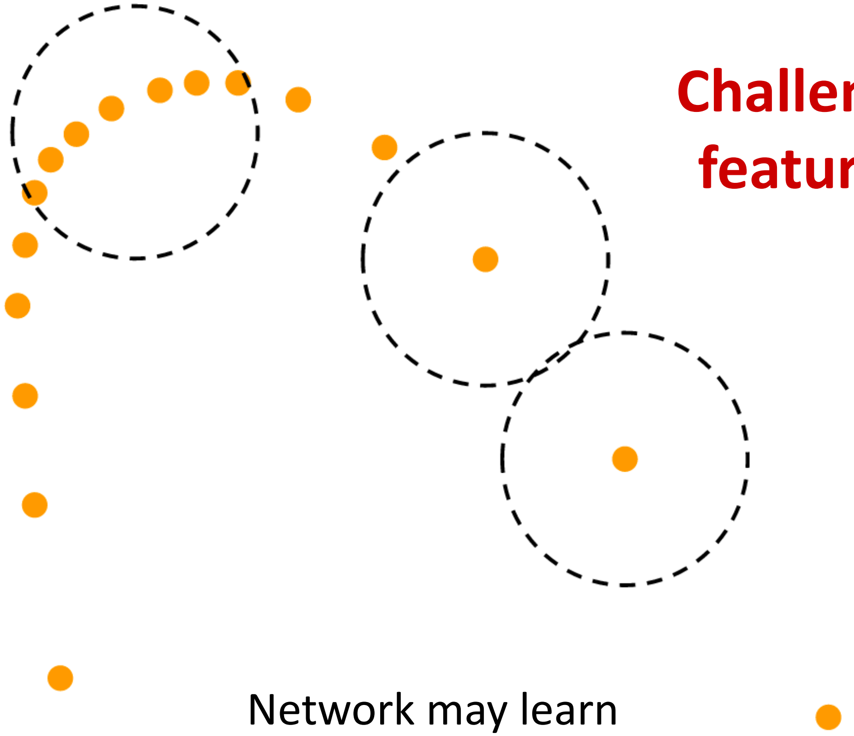
"Up-convolution" through 3D interpolation and/or pointnet.

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.

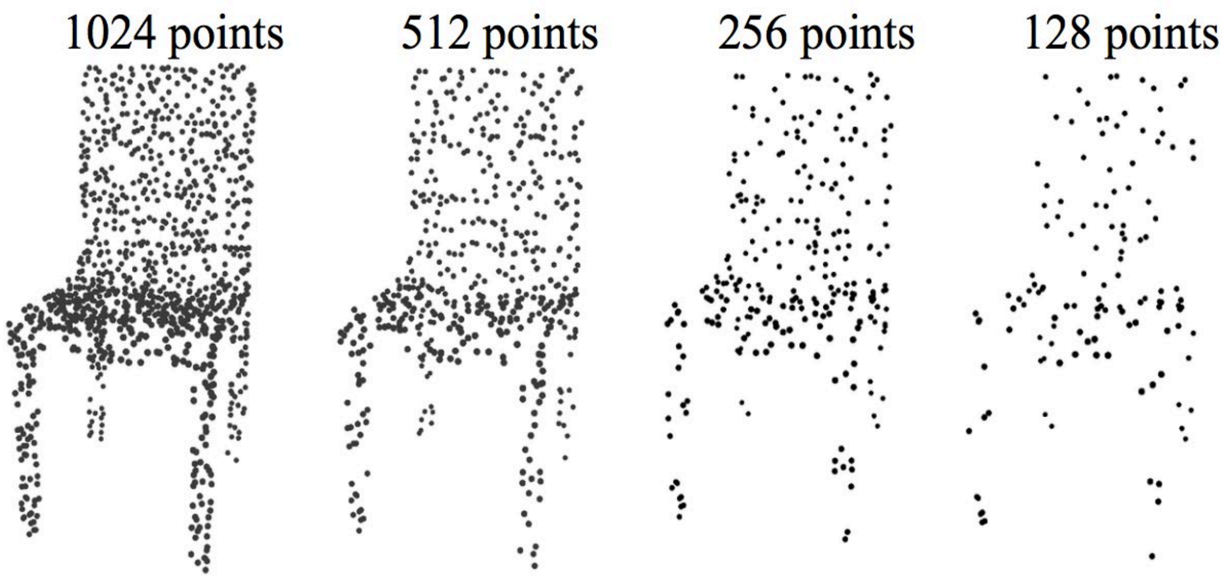


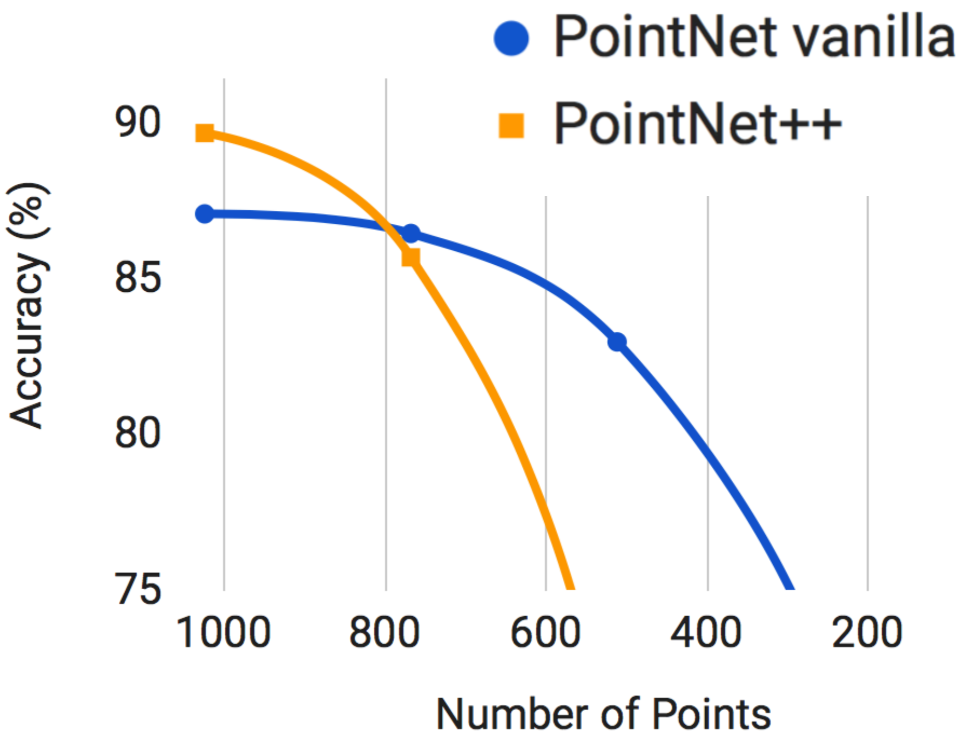


Challenge for local feature learning!

Network may learn sampling pattern – not underlying geometry

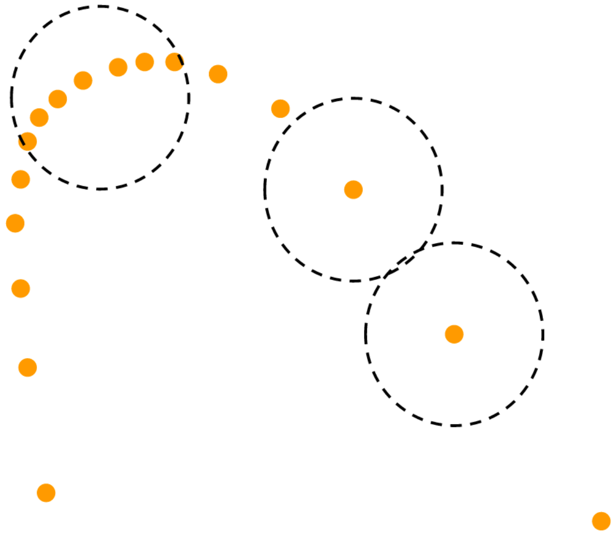
Density Variation Affects Hierarchy

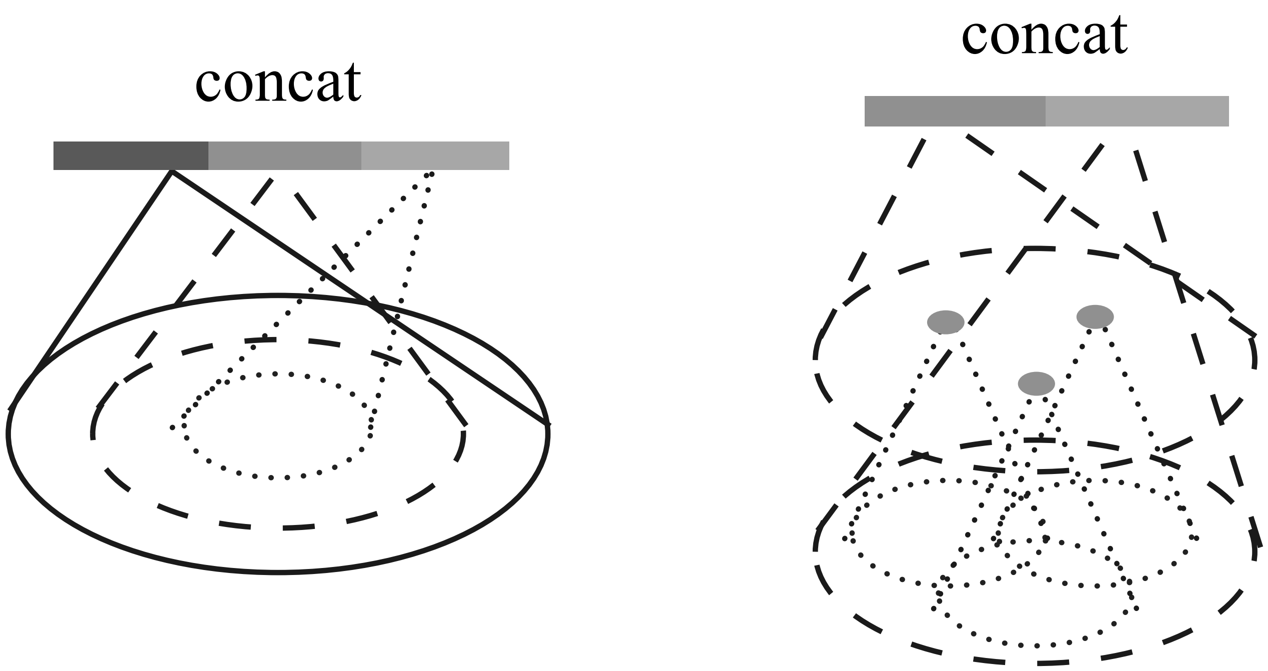




Small kernels suffer from varying densities!

Robust Learning Under Varying Sampling Density



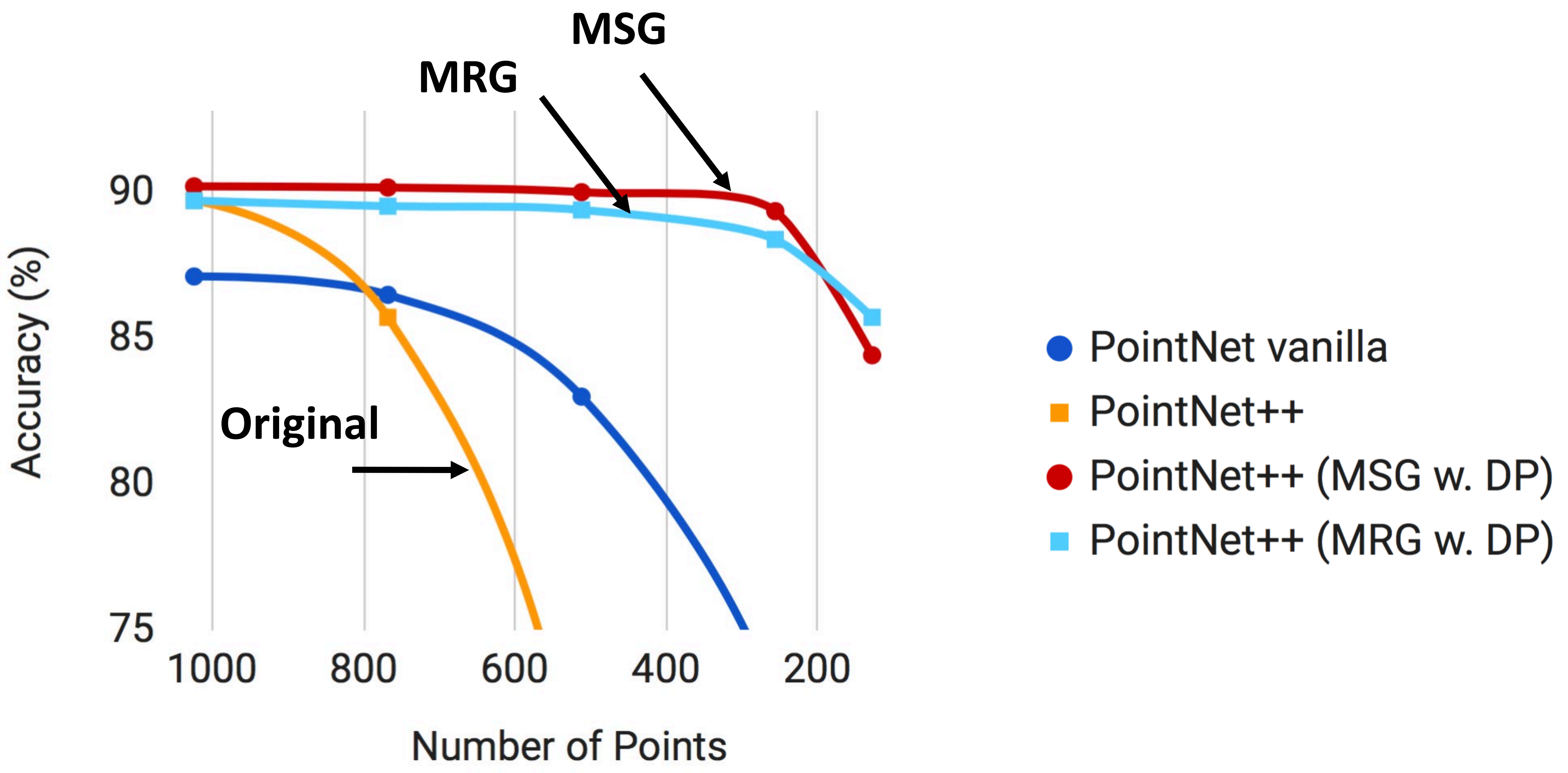


(a) (b)

Multi-scale grouping (MSG) Multi-res grouping (MRG)

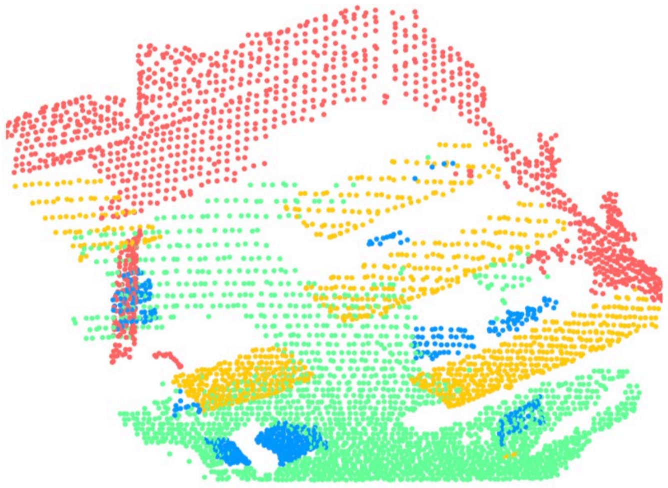
During Training: input point dropout with random dropout ratio

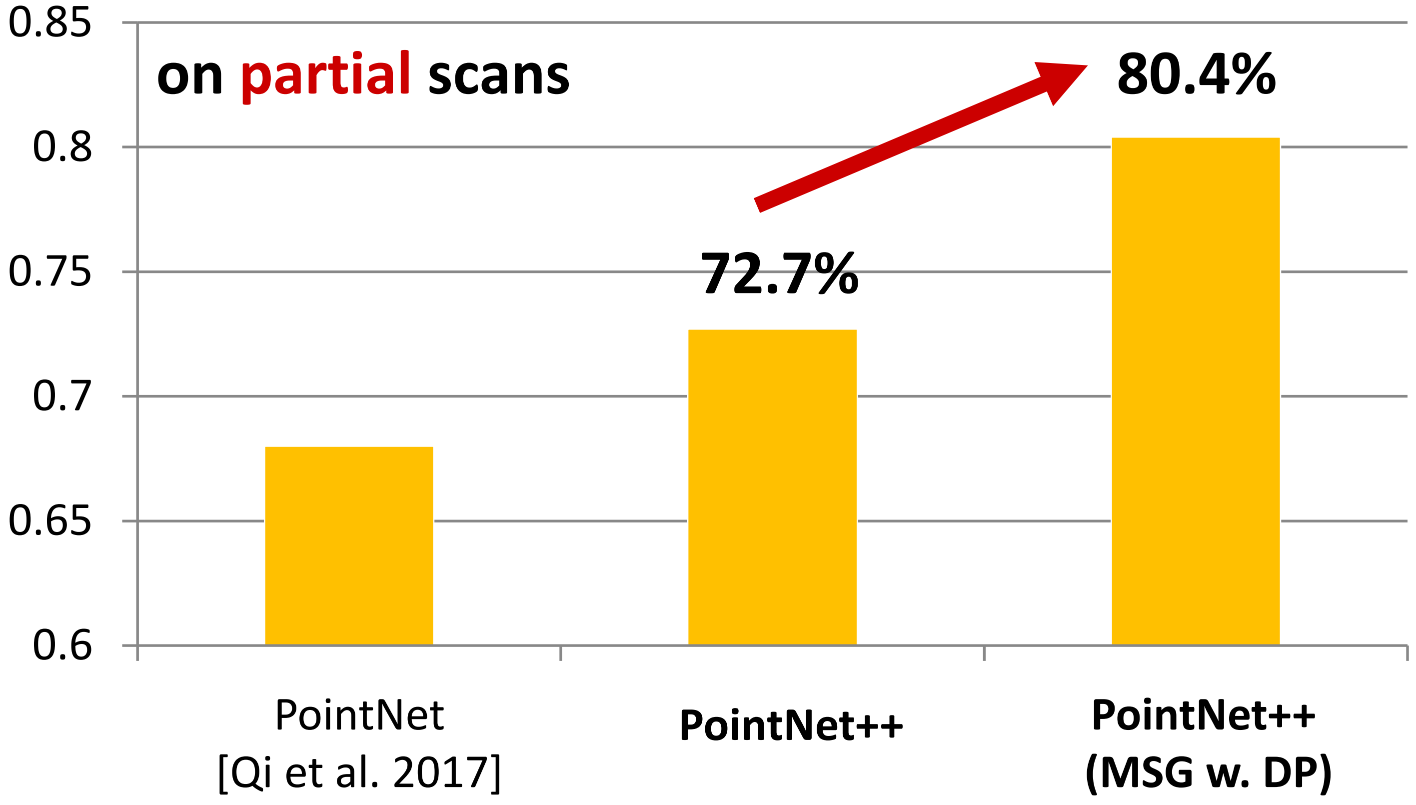
Robust Learning Under Varying Sampling Density



PointNet++ Results: Scene Parsing

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

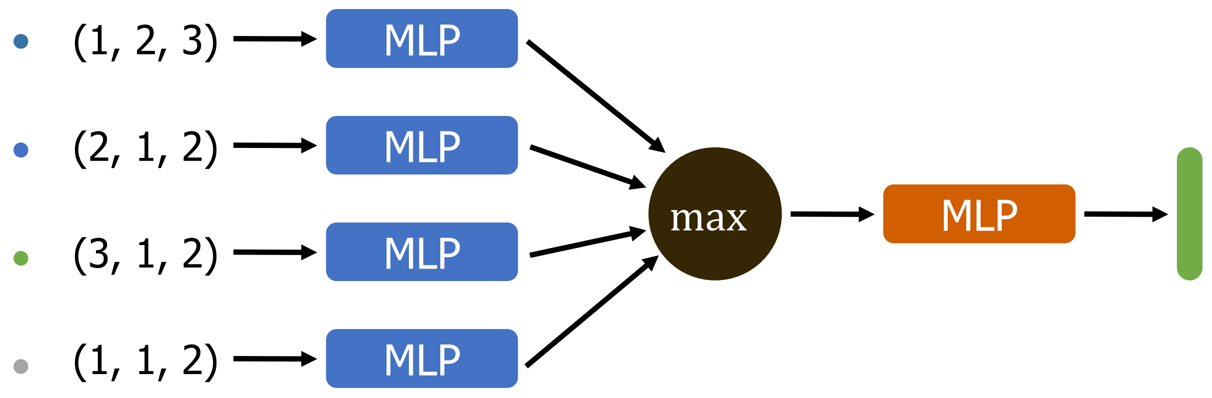




dataset: ScanNet; metric: per-point semantic classification accuracy (%)

Graph Structures on Points: DGCNN

Point Clouds and Graphs



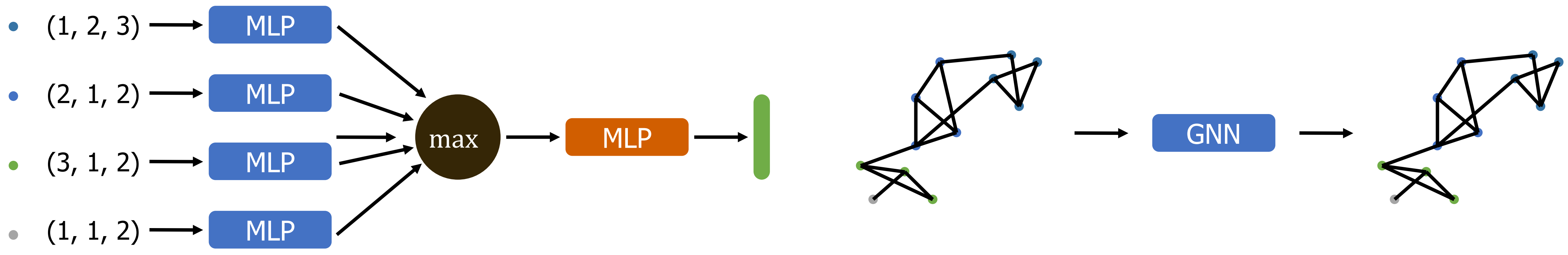
PointNet family

Graph Neural Networks

[Qi et al., CVPR 2017]

[Kipf et al., CVPR 2017]

Bridging the Two …. DGCNN



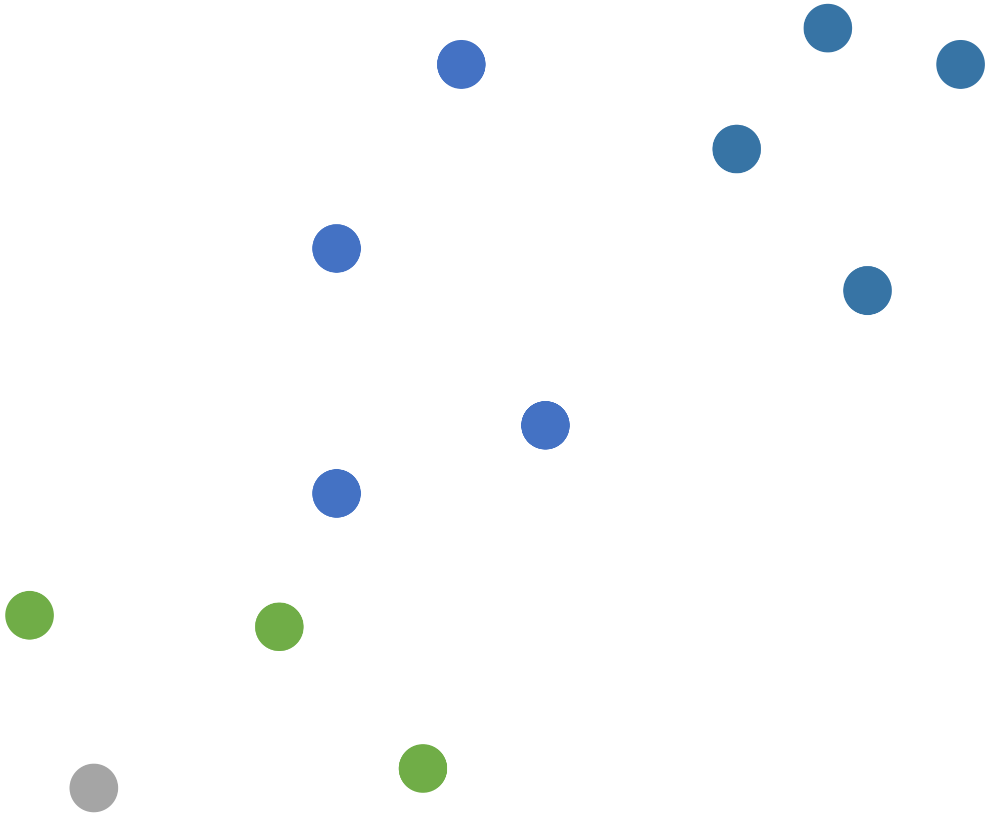
PointNet family

Dynamic Graph Convolutional Neural Networks (DGCNN)

Graph Neural Networks

[Dynamic Graph CNN for Learning on Point Clouds
Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, Justin M. Solomon, TOG 2019]

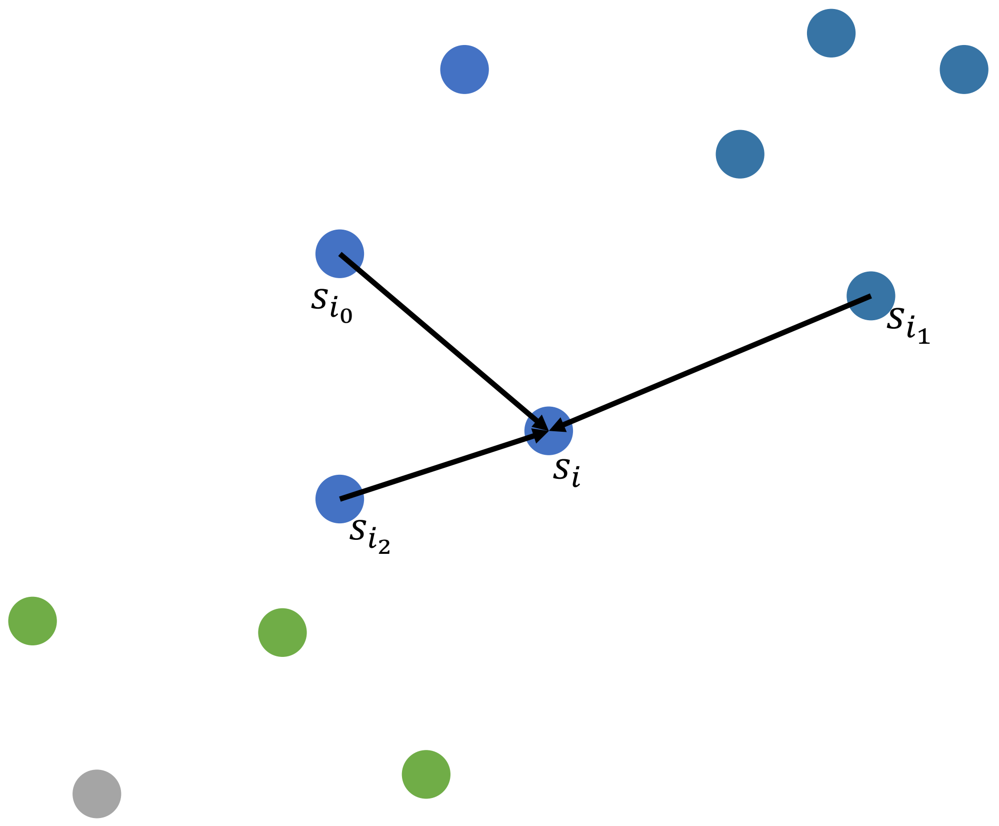
DGCNN



$$\mathcal{S} = \{(x_0, y_0, z_0), \ldots, (x_i, y_i, z_i), \ldots, (x_n, y_n, z_n)\}$$

[Wang et al., TOG 2019]

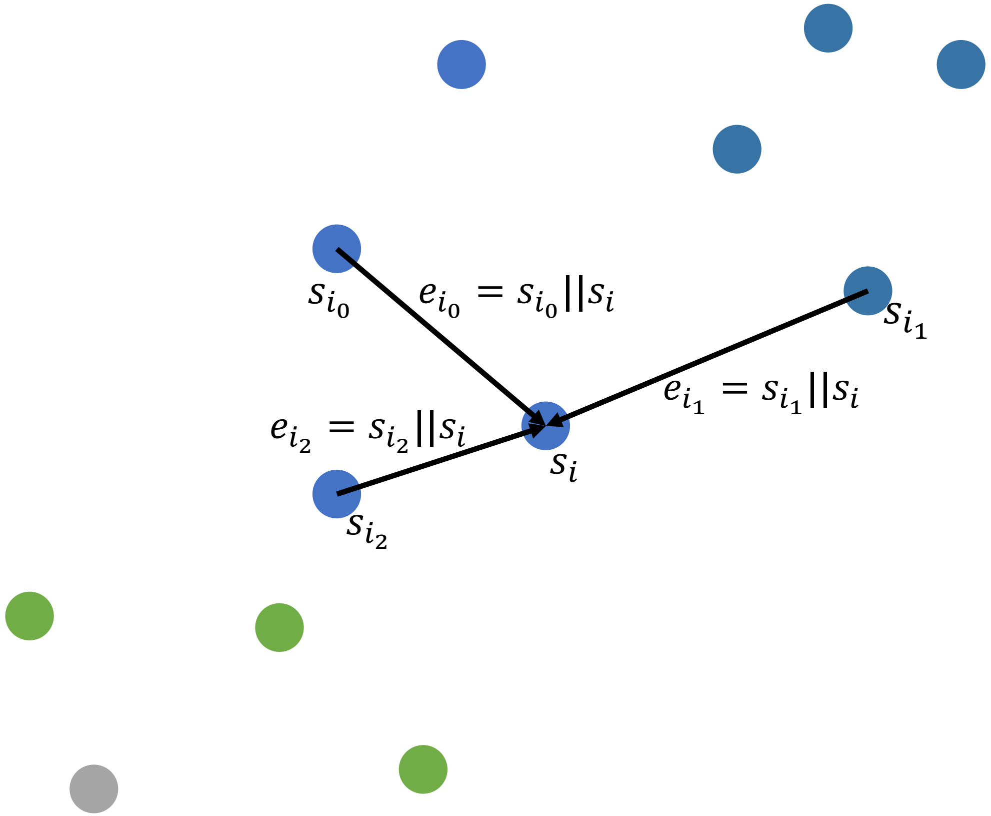
DGCNN



Each point (e.g., s_i) is connected to its k nearest neighbors

[Wang et al., TOG 2019]

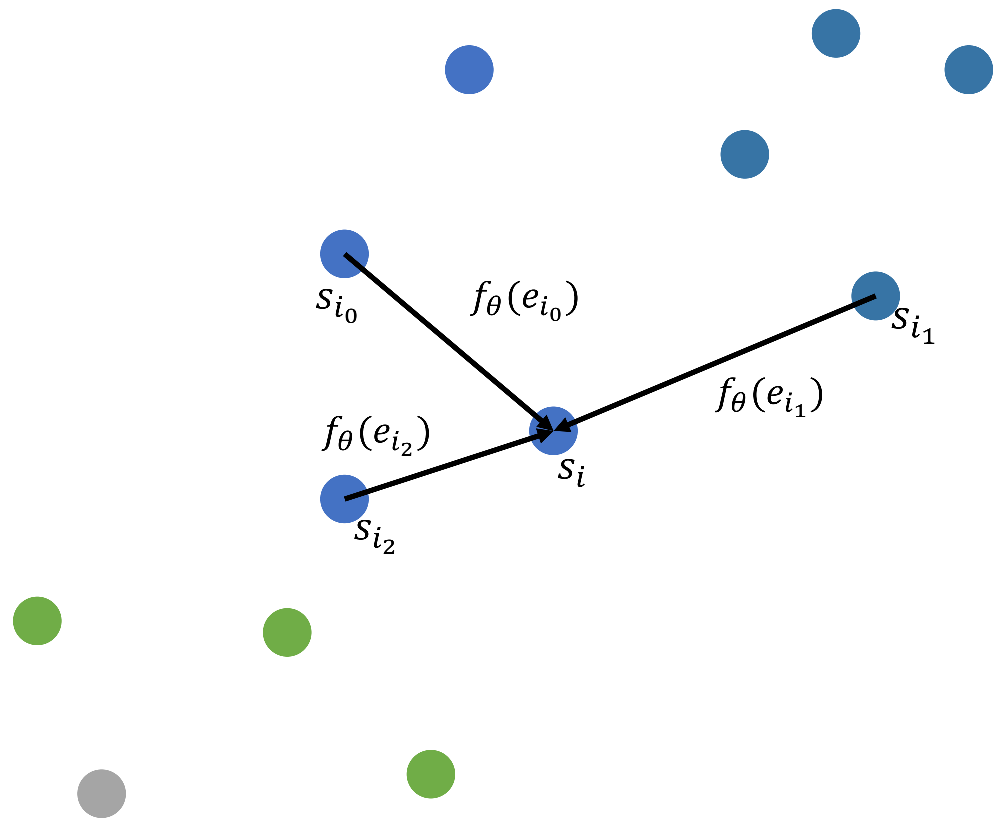
DGCNN



Edge features are defined by concatenating features of points

[Wang et al., TOG 2019]

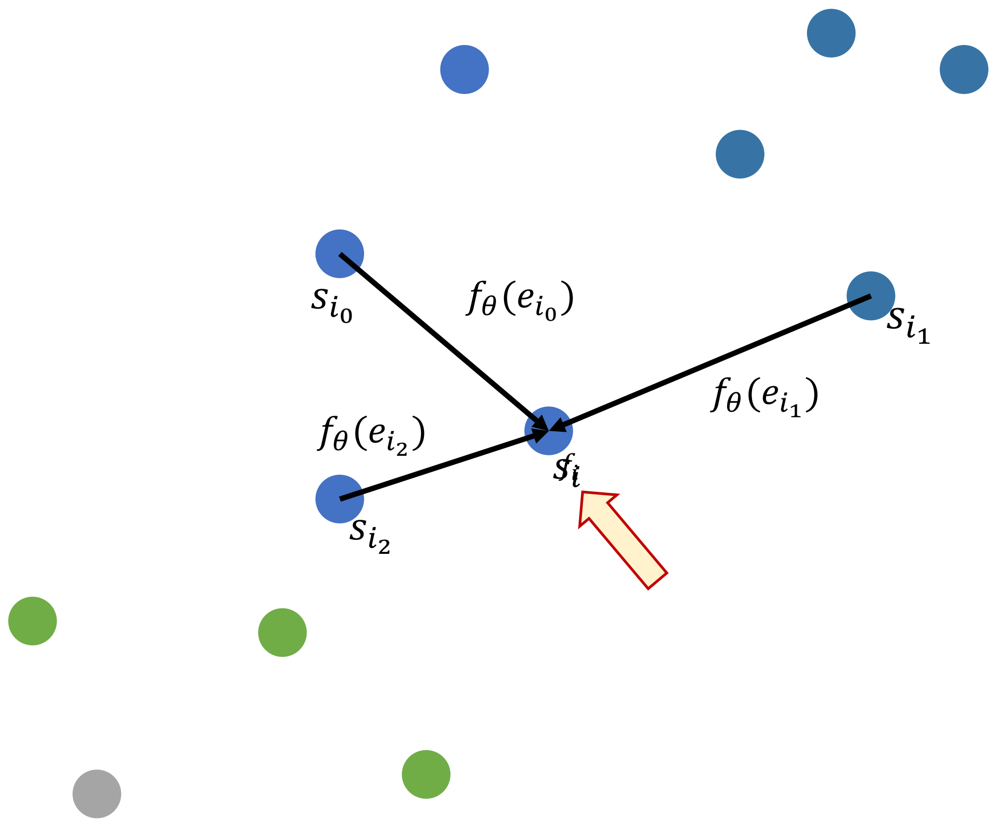
DGCNN



A shared MLP further lifts edge features into high dimensional space

[Wang et al., TOG 2019]

DGCNN – the EdgeConv Operation

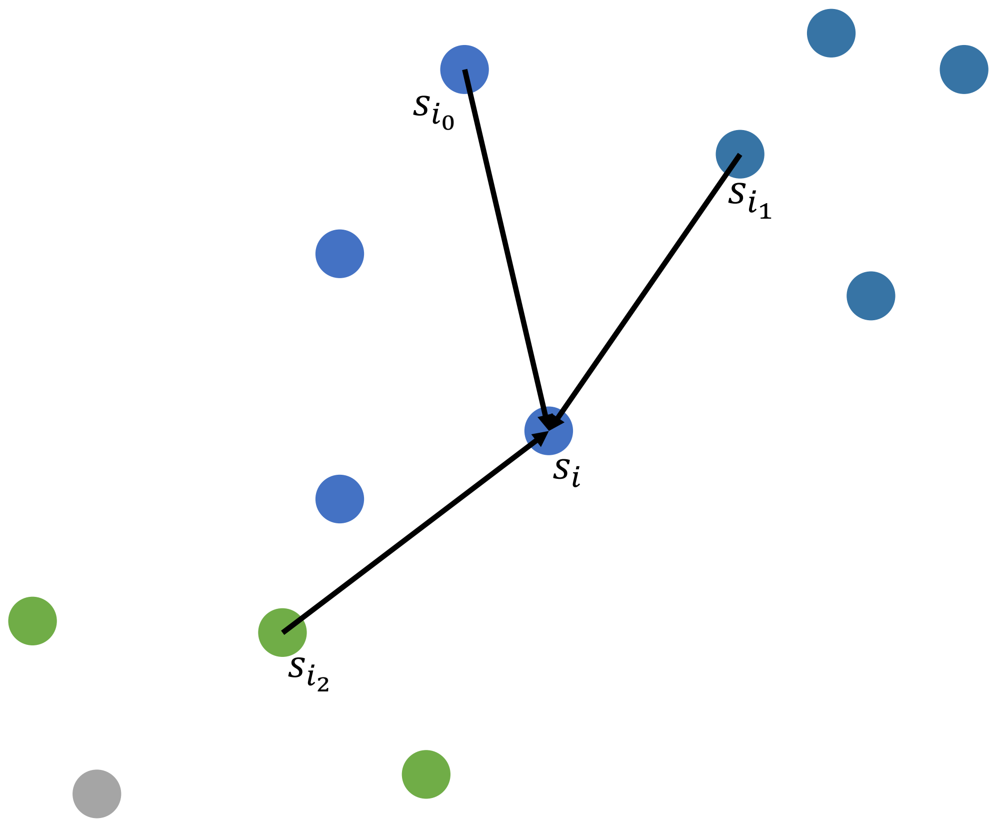


Features are pooled by a symmetric function

$$f_i = \max(f_\theta(e_{i_0}), f_\theta(e_{i_1}), f_\theta(e_{i_2}))$$

[Wang et al., TOG 2019]

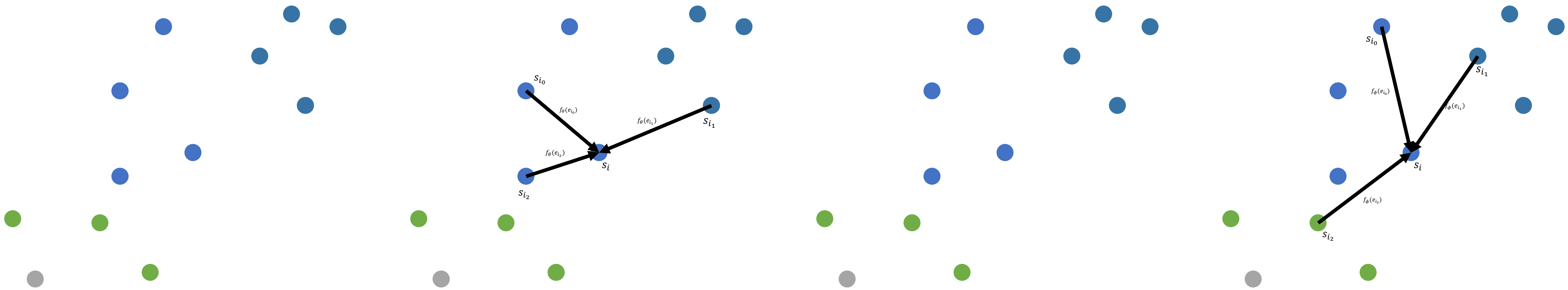
DGCNN



Then, a new kNN graph is reconstructed based on features.

[Wang et al., TOG 2019]

DGCNN – Alternating Processing



DGCNN alternates feature learning (EdgeConvs) and graph reconstruction

[Wang et al., TOG 2019]

EdgeConv Operation

edge features: $e_{ij} = h_\Theta(\mathbf{x}_i, \mathbf{x}_j)$

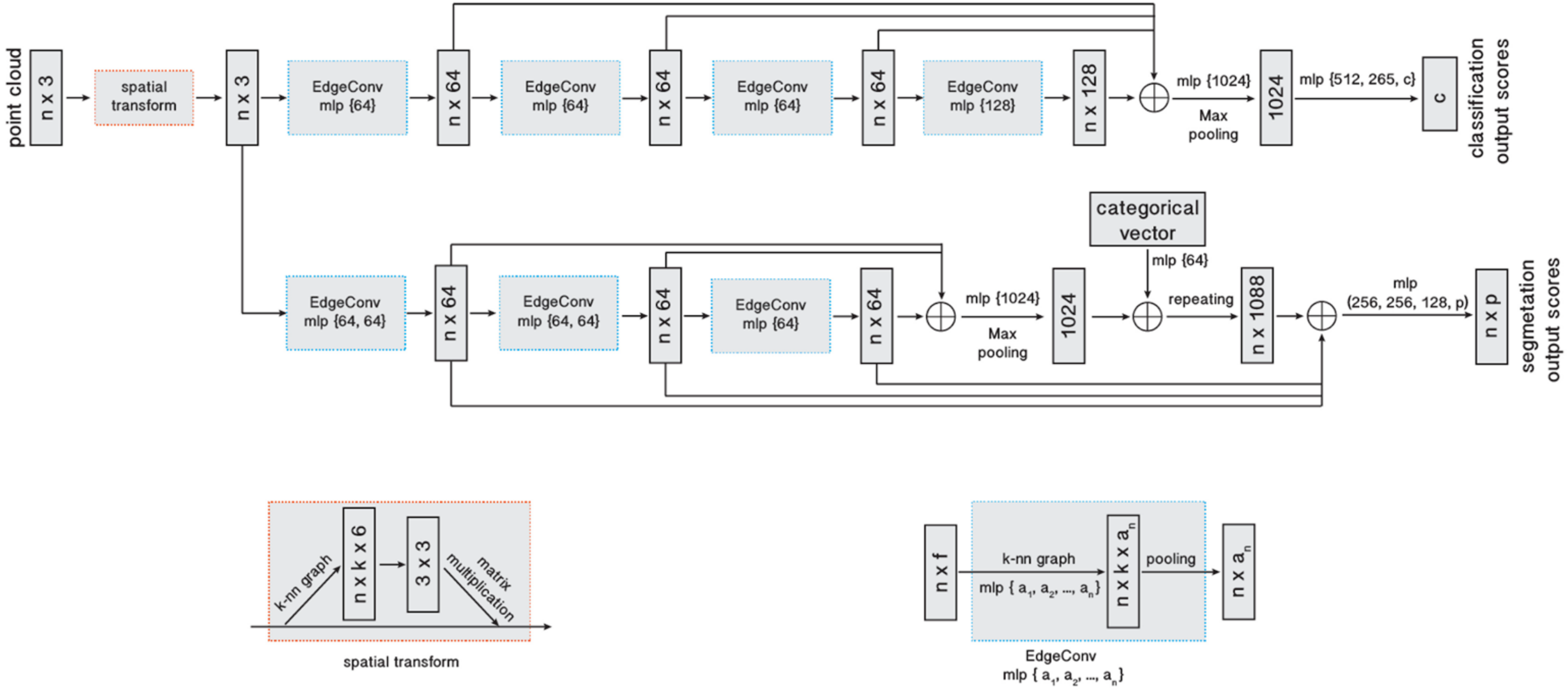
- general $\quad \mathbf{x}'_i = \underset{j:(i,j)\in\mathcal{E}}{\square} h_\Theta(\mathbf{x}_i, \mathbf{x}_j). \quad (1)$
- weighted sum $\quad x'_{im} = \sum_{j:(i,j)\in\mathcal{E}} \theta_m \cdot \mathbf{x}_j. \quad (2)$
- global only $\quad h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = h_\Theta(\mathbf{x}_i), \quad (3)$
- local only $\quad h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = h_\Theta(\mathbf{x}_j - \mathbf{x}_i). \quad (6)$
- global + local $\quad h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = \bar{h}_\Theta(\mathbf{x}_i, \mathbf{x}_j - \mathbf{x}_i). \quad (7)$

Key EdgeConv Properties

- Easily implement and integrates into existing deep learning-based algorithm by switching the MLP component to EdgeConv
- EdgeConv is differentiable, which is an important property in ML and DL (convex optimization problem)
- Extract local features without destroying the permutation invariance
- Dynamic Graph CNN => update the graph after each layer of network, i.e. recompute the k-nearest neighbors in the new feature space (and also recompute the edge features)

DGCNN Architecture

DGCNN



DGCNN Results

	Mean Class Accuracy	Overall Accuracy
3DShapeNets [Wu et al. 2015]	77.3	84.7
VoxNet [Maturana and Scherer 2015]	83.0	85.9
Subvolume [Qi et al. 2016]	86.0	89.2
VRN (single view) [Brock et al. 2016]	88.98	-
VRN (multiple views) [Brock et al. 2016]	91.33	-
ECC [Simonovsky and Komodakis 2017]	83.2	87.4
PointNet [Qi et al. 2017b]	86.0	89.2
PointNet++ [Qi et al. 2017c]	-	90.7
Kd-net [Klokov and Lempitsky 2017]	-	90.6
PointCNN [Li et al. 2018a]	88.1	92.2
PCNN [Atzmon et al. 2018]	-	92.3
Ours (baseline)	88.9	91.7
Ours	**90.2**	**92.9**
Ours (2048 points)	**90.7**	**93.5**

Table 2. Classification results on ModelNet40.

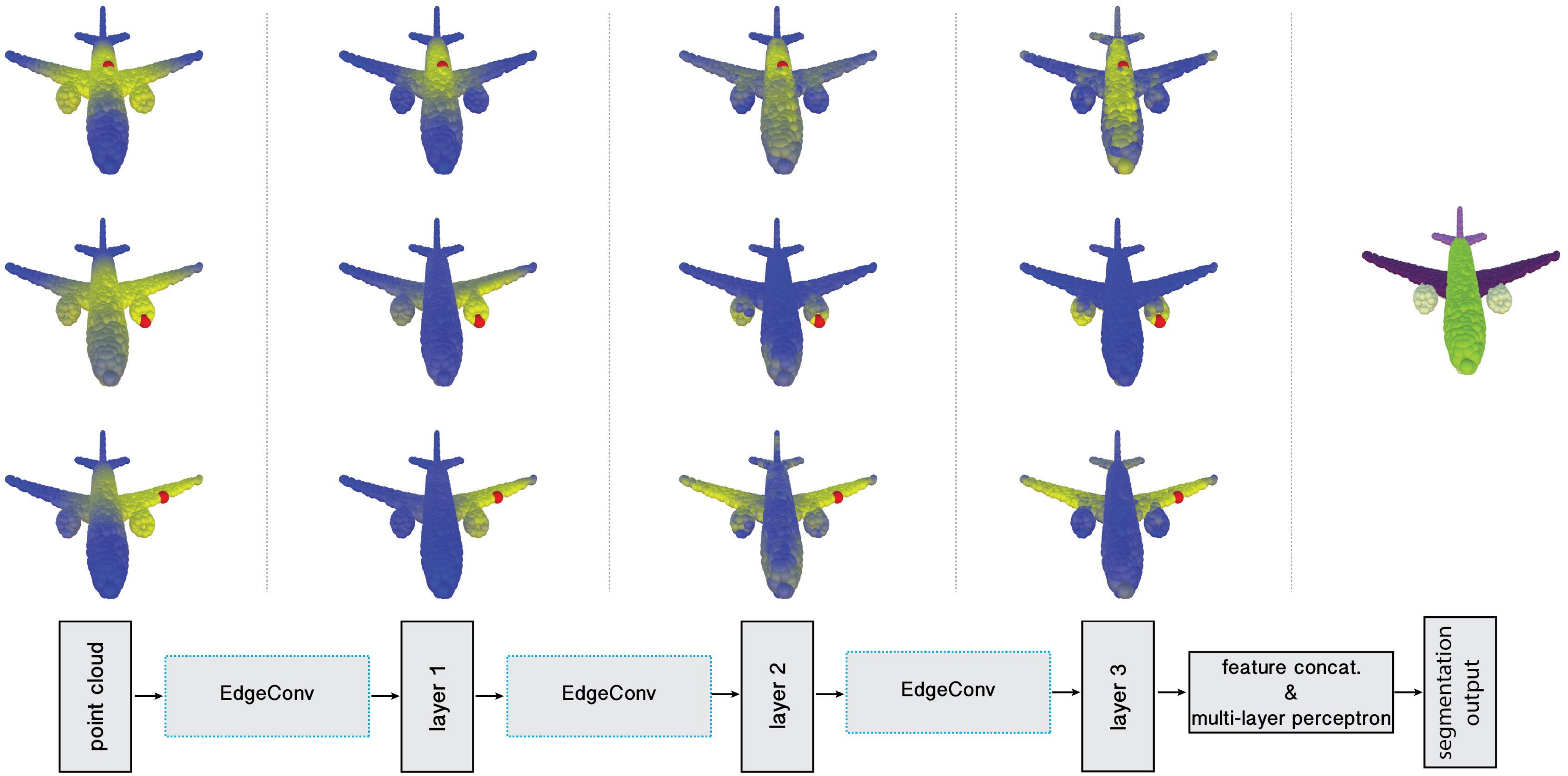
	Model size(MB)	Time(ms)	Accuracy(%)
PointNet (Baseline) [Qi et al. 2017b]	9.4	6.8	87.1
PointNet [Qi et al. 2017b]	40	16.6	89.2
PointNet++ [Qi et al. 2017c]	12	163.2	90.7
PCNN [Atzmon et al. 2018]	94	117.0	92.3
Ours (Baseline)	11	19.7	91.7
Ours	21	27.2	92.9

Table 3. Complexity, forward time, and accuracy of different models

DGCNN achieves superior performance on shape classification tasks while maintaining simplicity!

[Wang et al., TOG 2019]

From Geometry to Semantics



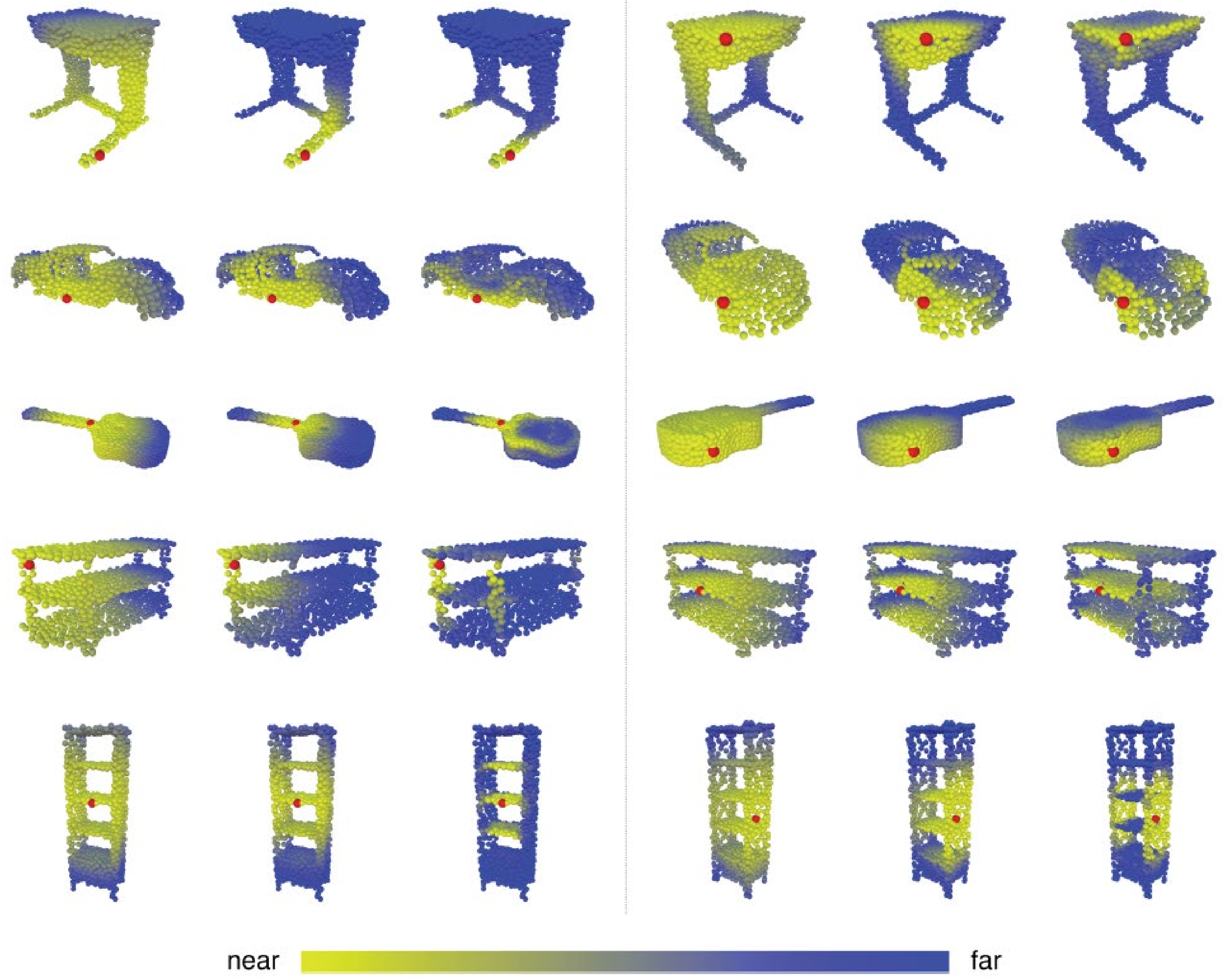
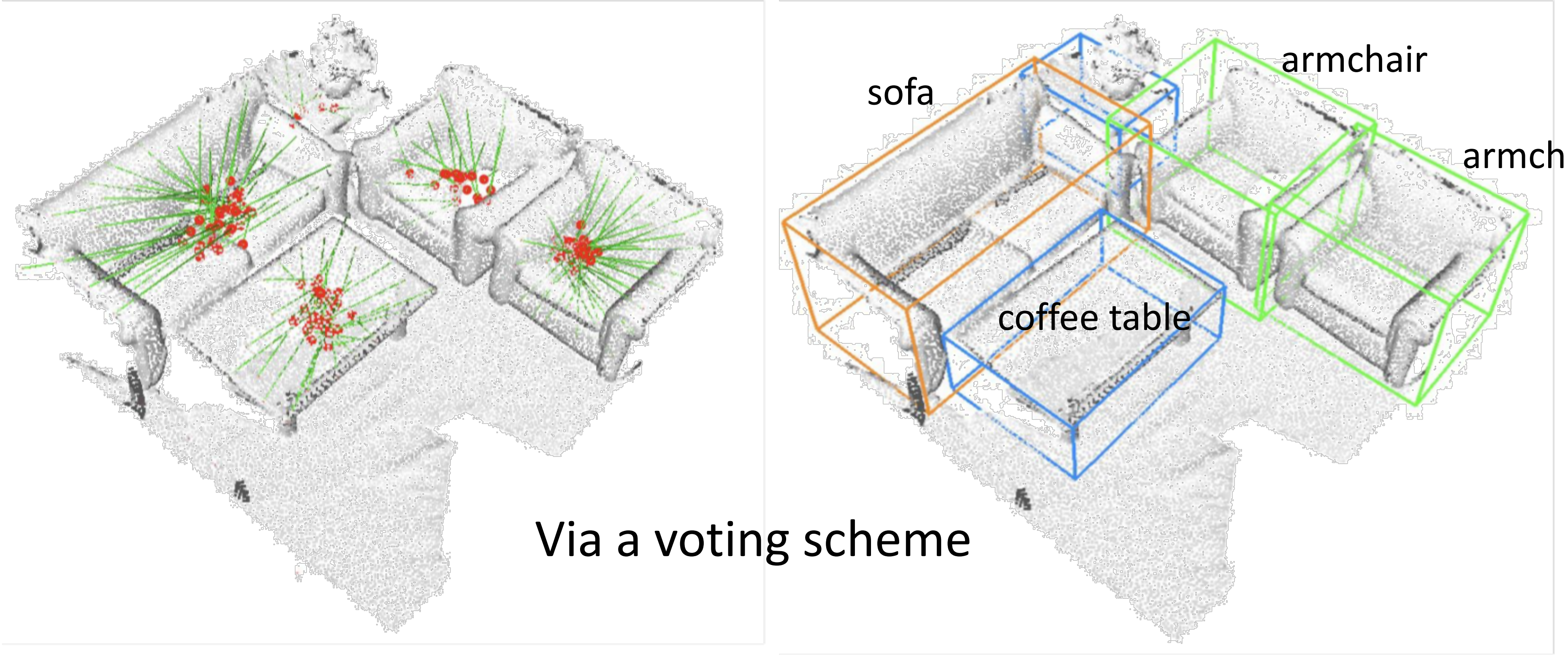


Fig. 4. Structure of the feature spaces produced at different stages of our shape classification neural network architecture, visualized as the distance between the red point to the rest of the points. For each set, Left: Euclidean distance in the input $\mathbb{R}^3$ space; Middle: Distance after the point cloud transform stage, amounting to a global transformation of the shape; Right: Distance in the feature space of the last layer. Observe how in the feature space of deeper layers semantically similar structures such as shelves of a bookshelf or legs of a table are brought close together, although they are distant in the original space.

Object Detection in Point Clouds, Indoors

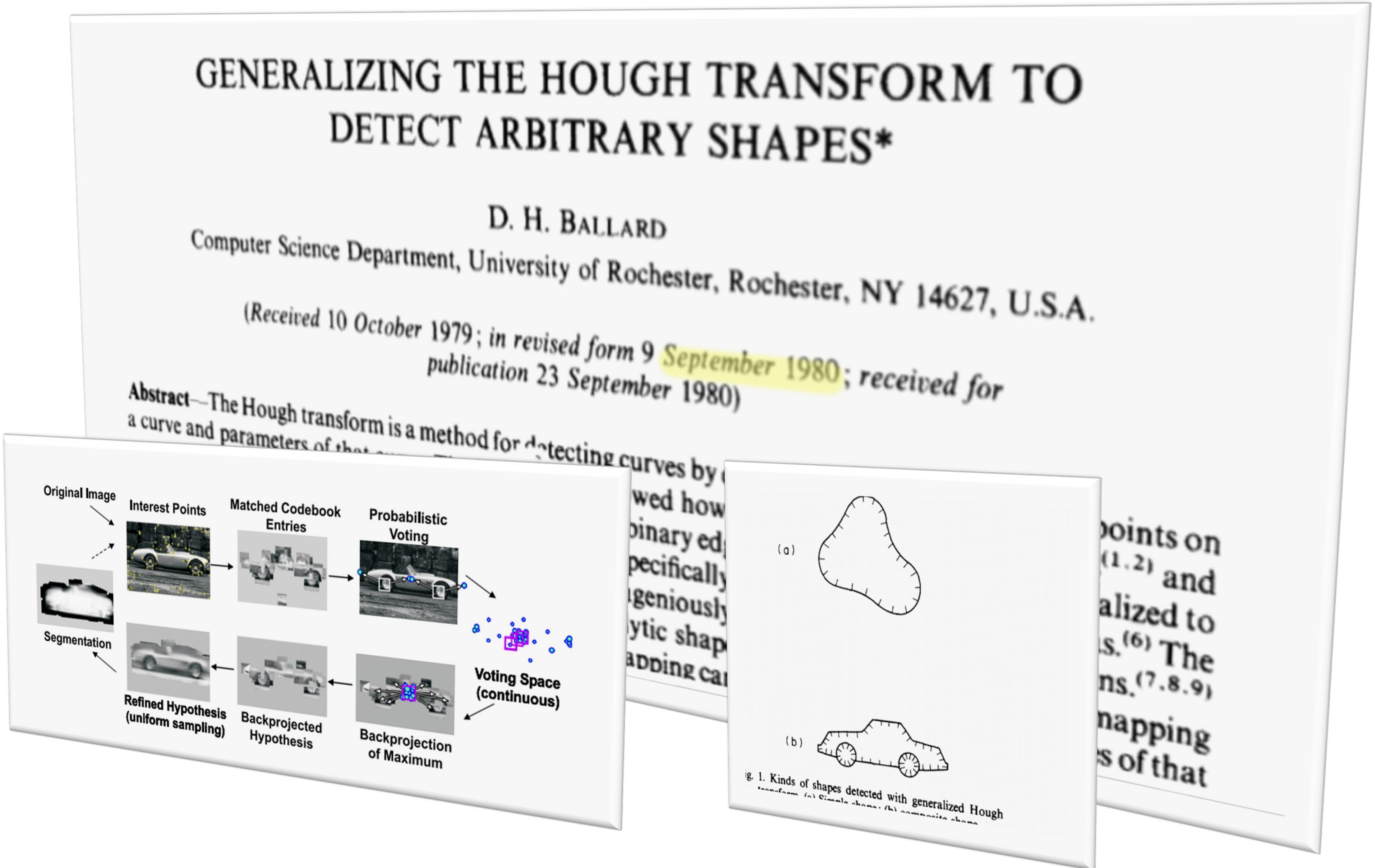
Point Cloud Object Amodal Bounding Box Detection



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 2019.
- Charles R. Qi, Xinlei Chen, Or Litany, Leonidas J. Guibas. *ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes*. CVPR 2020.

Generalized Hough Transform



GENERALIZING THE HOUGH TRANSFORM TO DETECT ARBITRARY SHAPES*

D. H. BALLARD

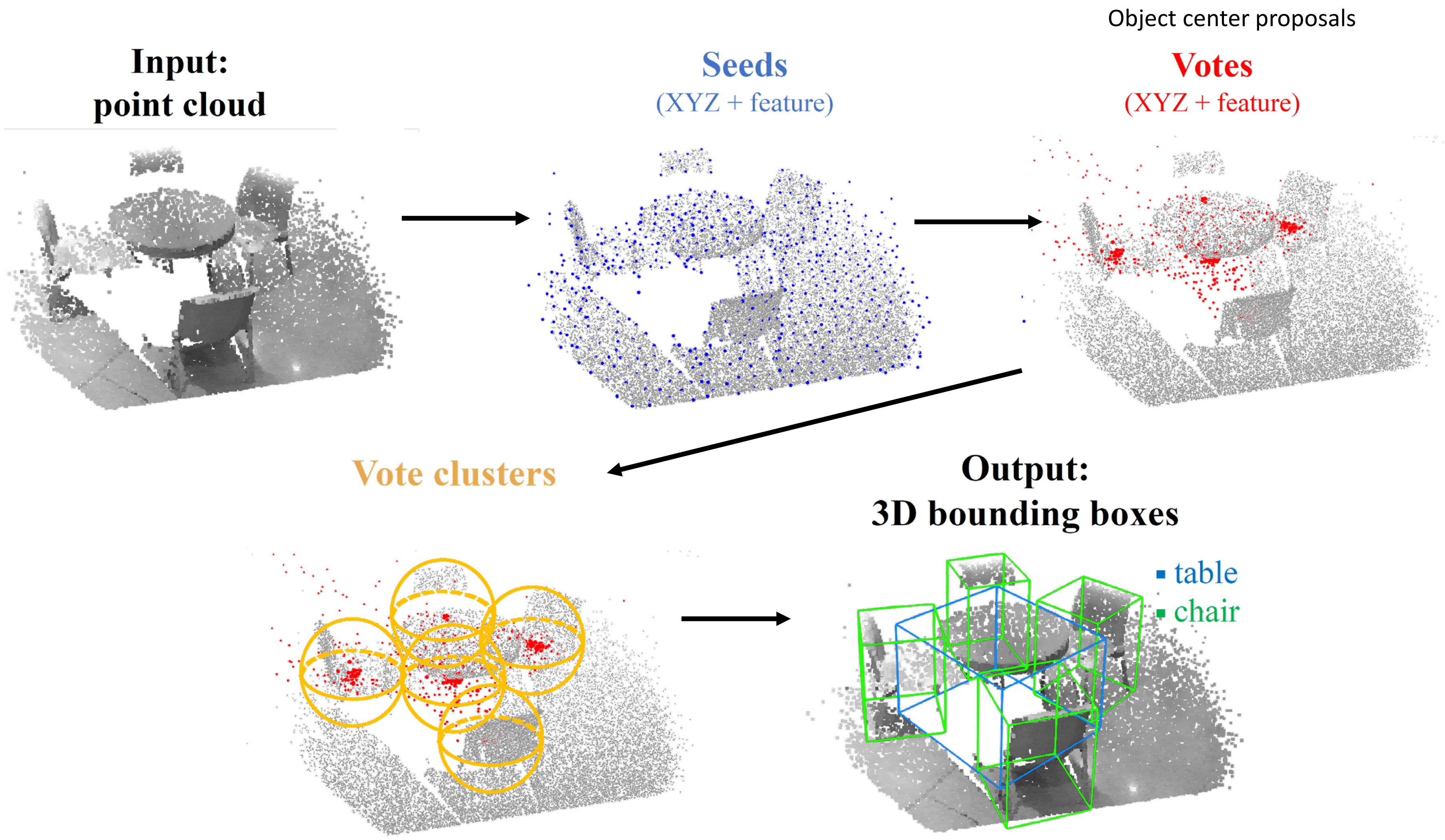
Computer Science Department, University of Rochester, Rochester, NY 14627, U.S.A.

(Received 10 October 1979; in revised form 9 September 1980; received for publication 23 September 1980)

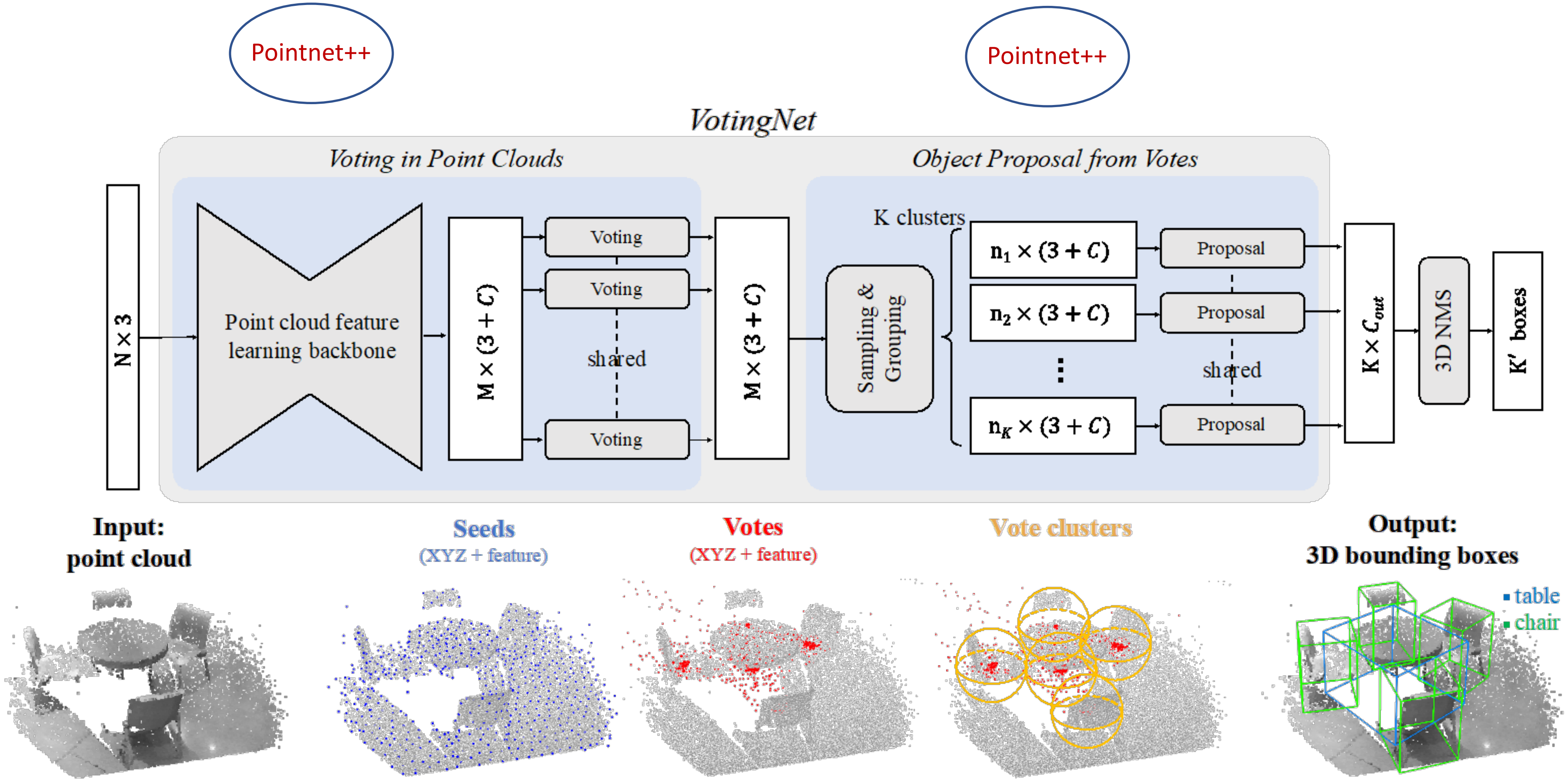
Abstract—The Hough transform is a method for detecting curves by
a curve and parameters of that

ig. 1. Kinds of shapes detected with generalized Hough

Deep Hough Voting – A Two-Stage Approach

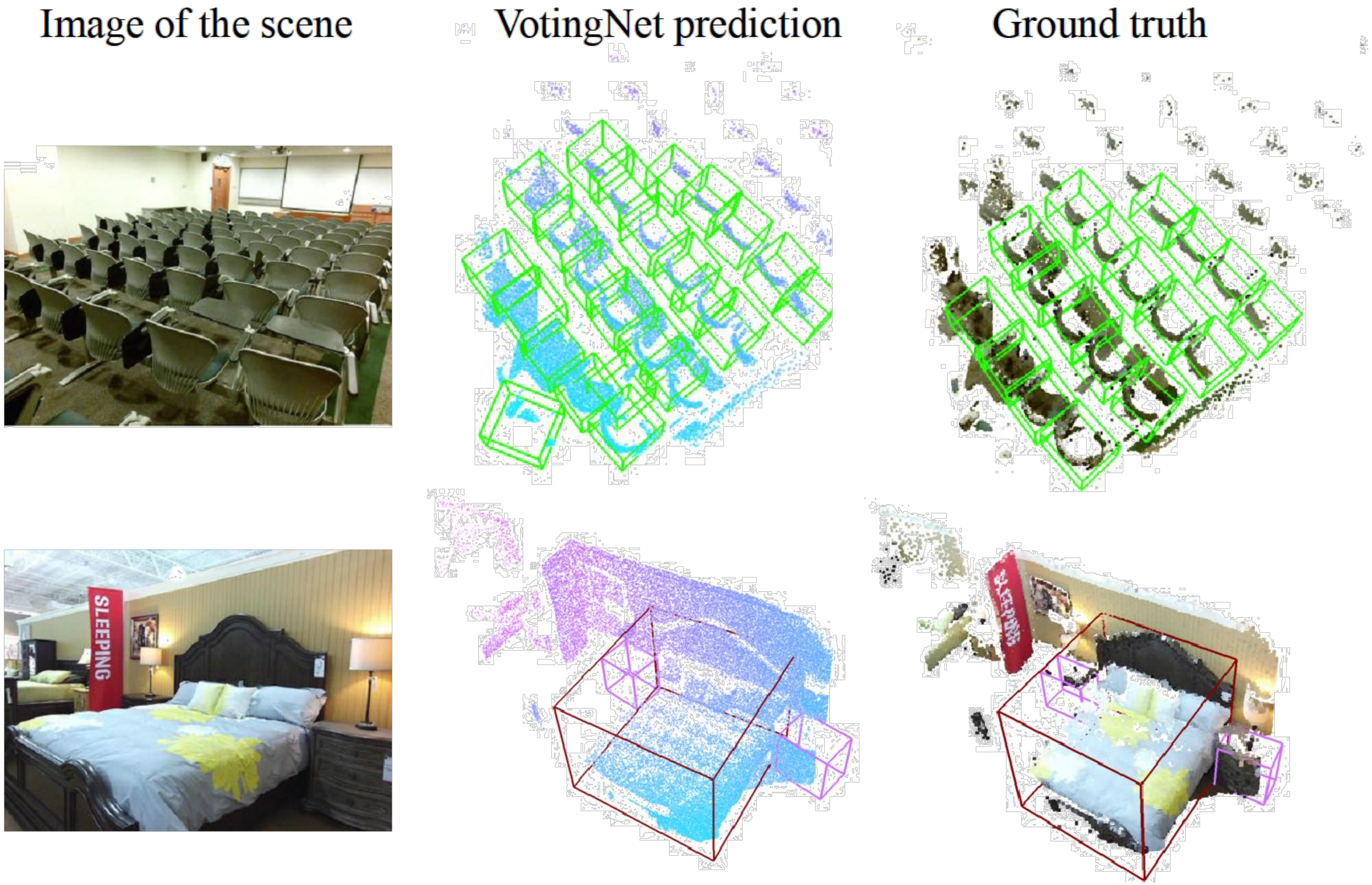


VoteNet – A Two-Stage Approach



A capsule/transformer network in disguise …

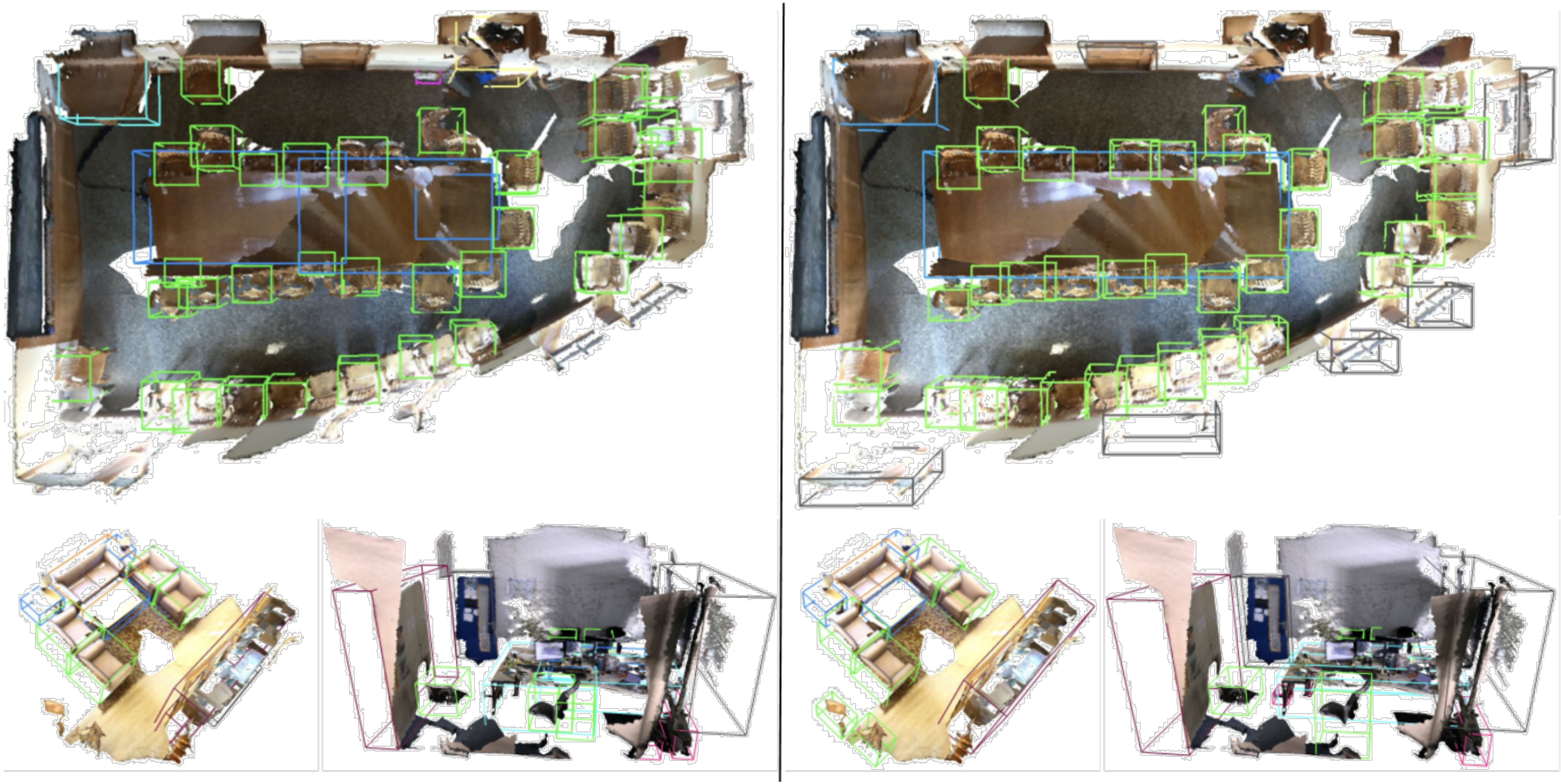
VoteNet Results on SUN RGB-D



VoteNet Results on ScanNet

VotingNet prediction

Ground truth



VoteNet Quantitative Results

average precision with 3D IoU threshold 0.25

SUN RGB-D

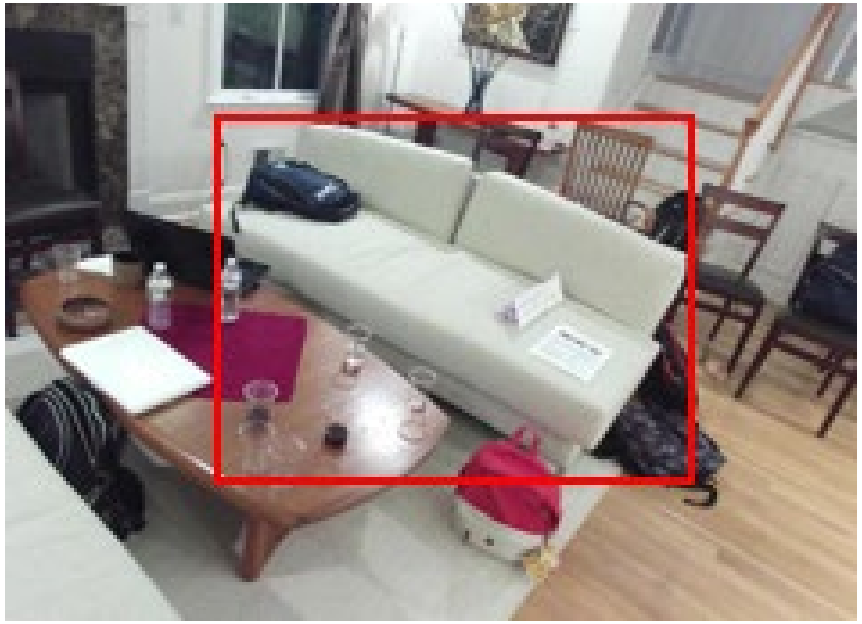
Deep sliding shapes
Clouds of oriented gradients
Frustum pointnet

	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**

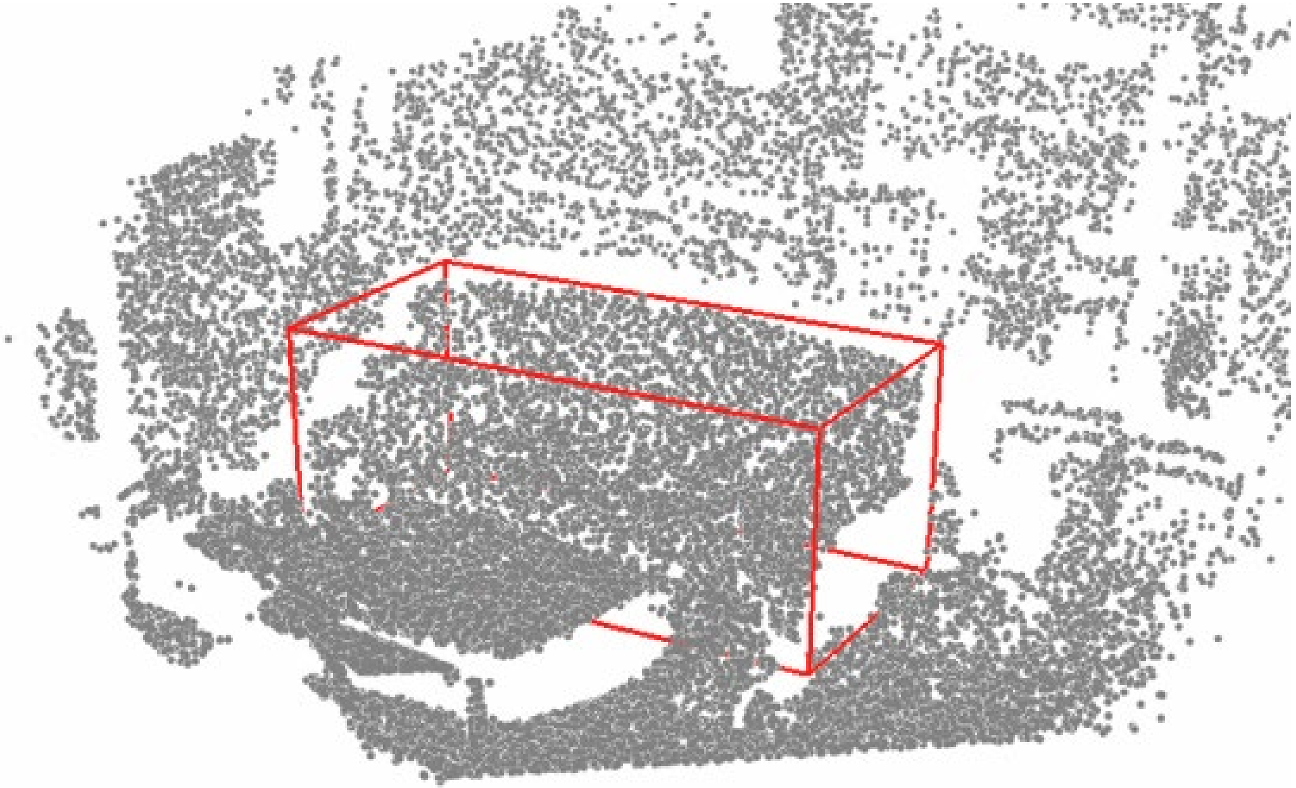
ScanNetV2

	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**

Modalities: Point Clouds Complement RGB Images



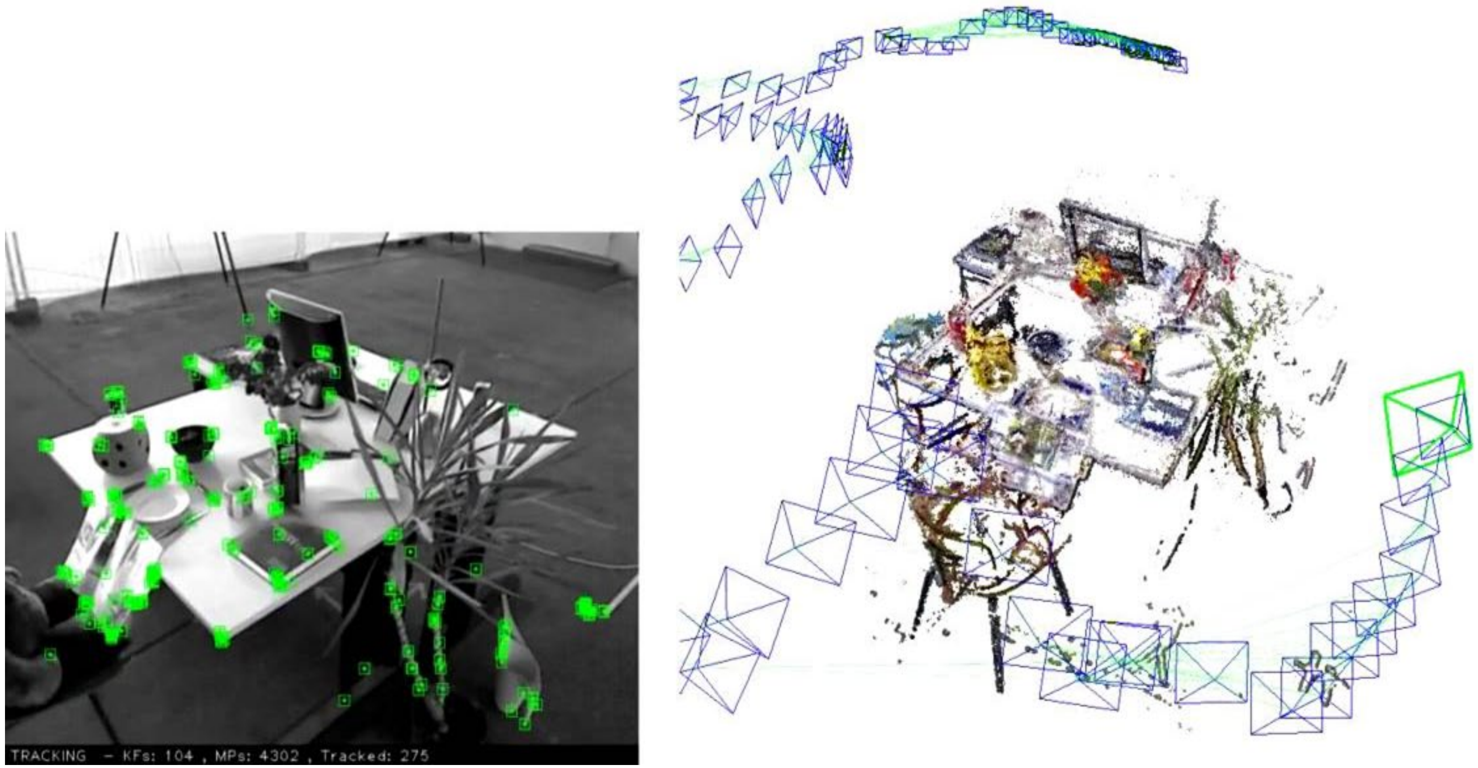
- + High resolution
- + Dense coverage
- - Subject to many imaging artifacts



- + Absolute depth and scale
- - Sparse, low rez

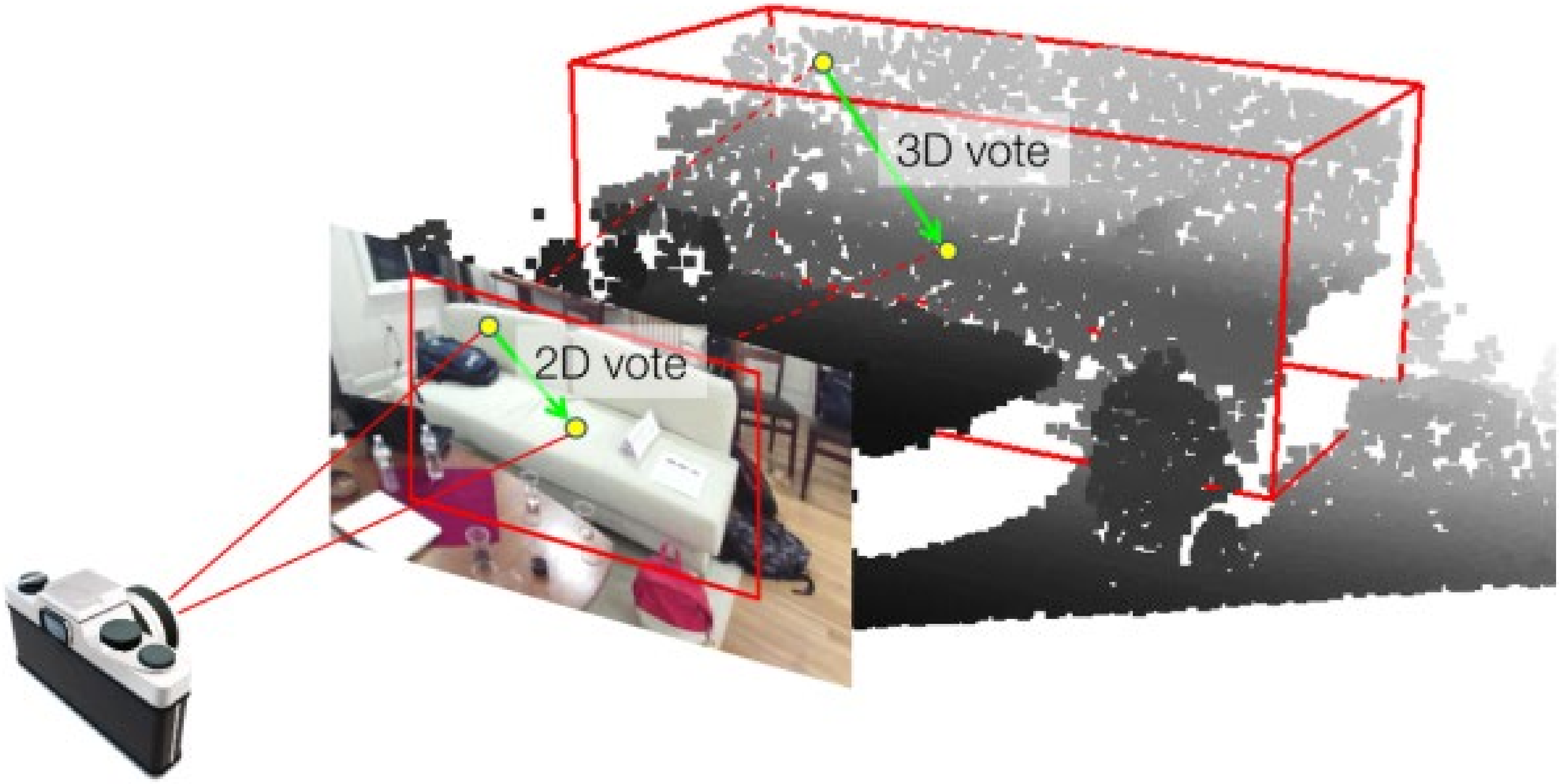
3D Detection with Sparse Points

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



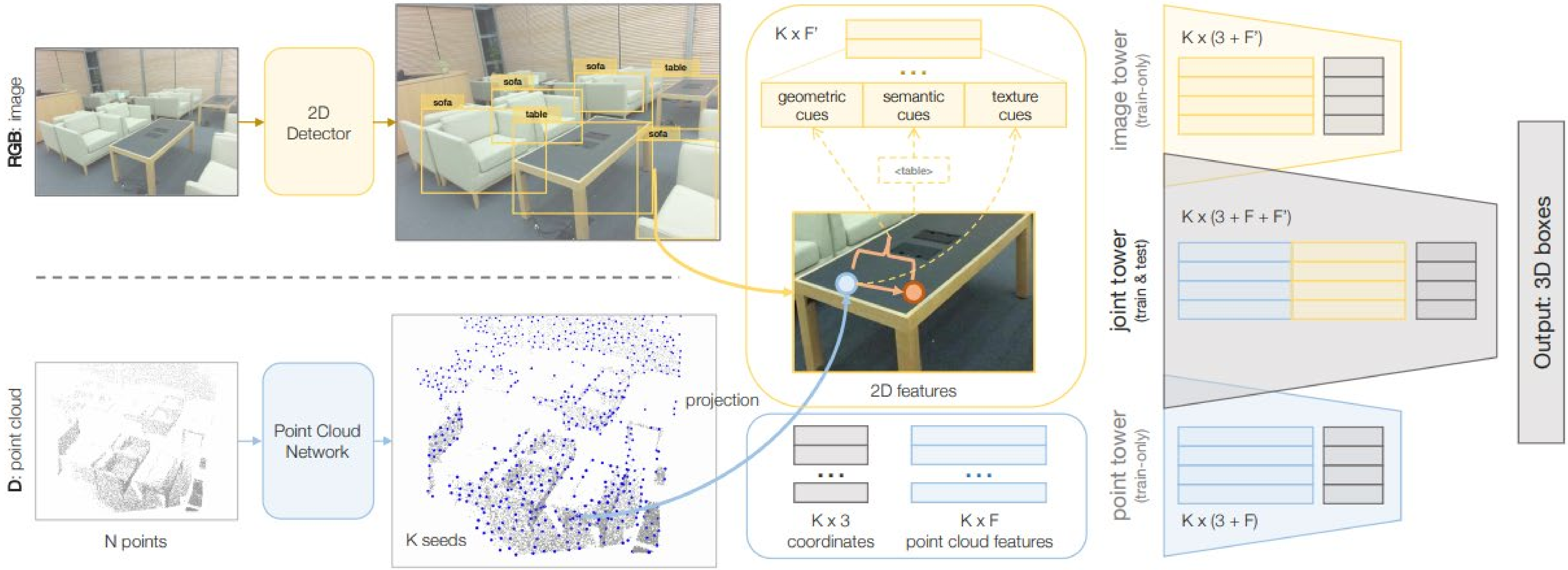
Picture: ORB-SLAM results

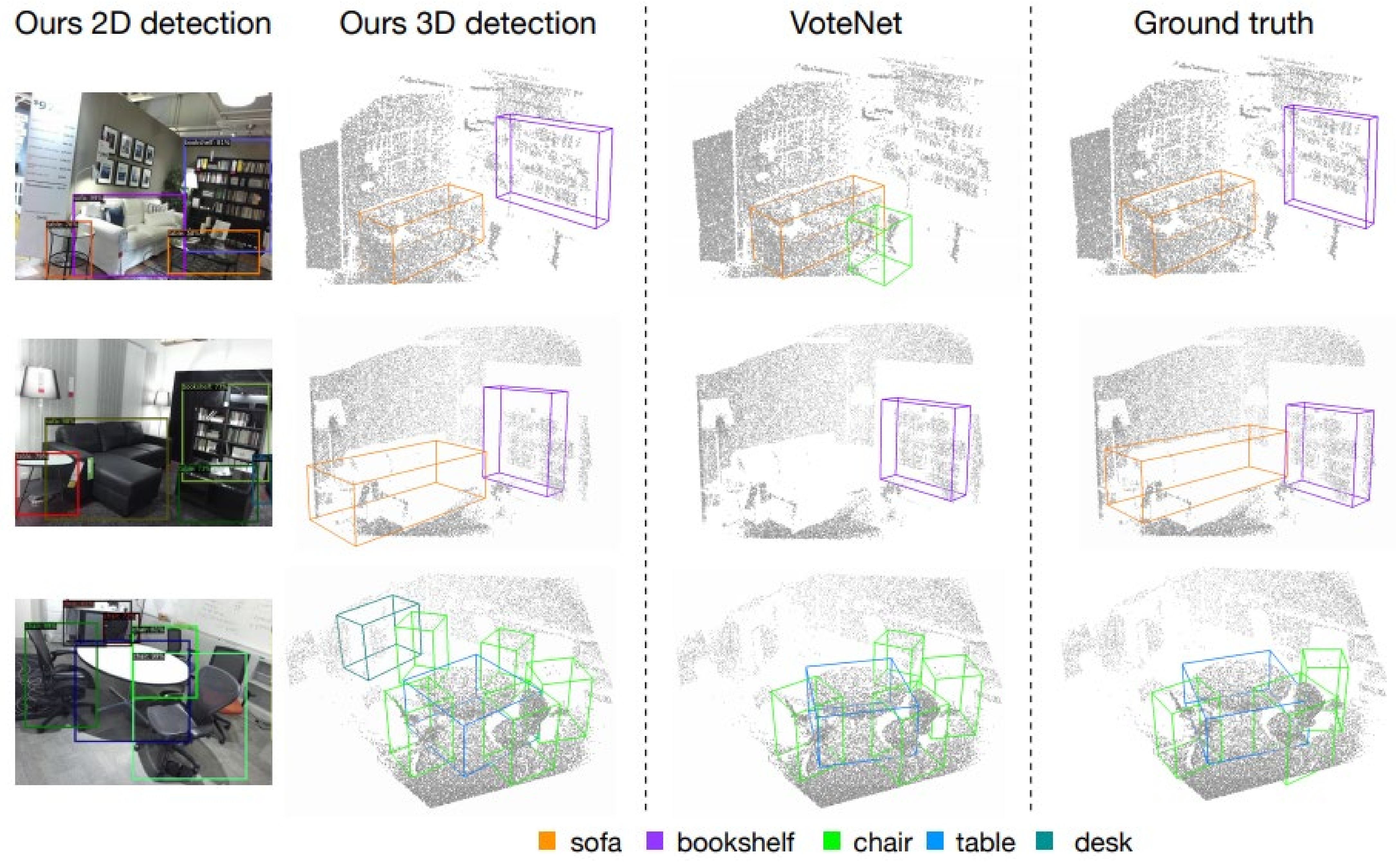
Points and Images: ImVoteNet



How to best combine geometry and appearance info?

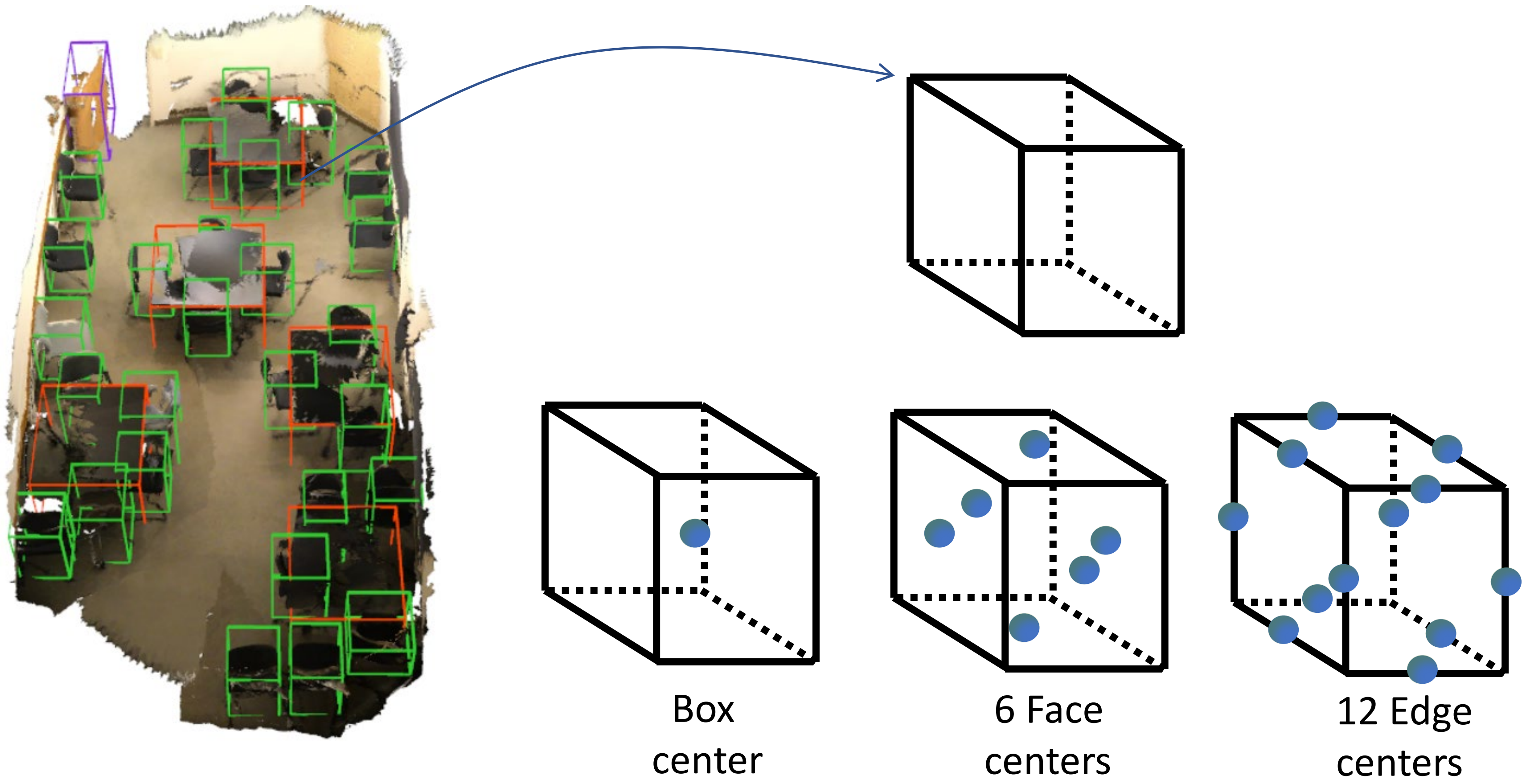
ImVoteNet Architecture





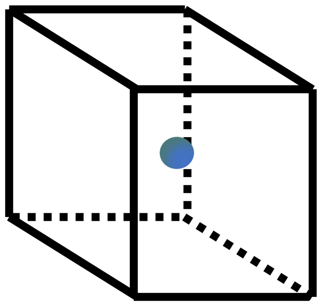
Ours 2D detection
Ours 3D detection
VoteNet
Ground truth
sofa
bookshelf
chair
table
desk

Multiple 3D Geometry Representations

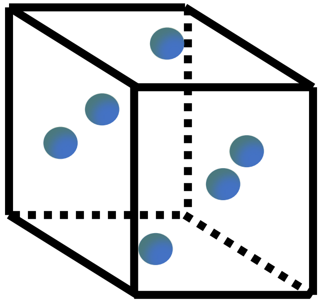


[Z. Zhang, B. Sun, H. Yang, Q. Huang.
H3DNet: 3D Object Detection Using Hybrid Geometric Primitives.
ECCV 2020]

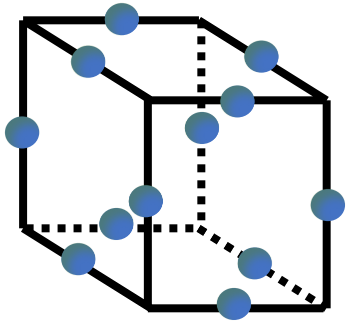
Representations Best for Different Object Instances



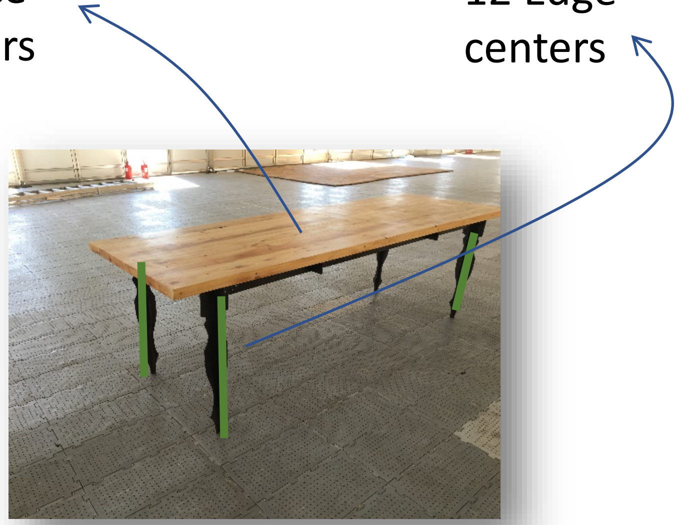
Box
center



6 Face
centers

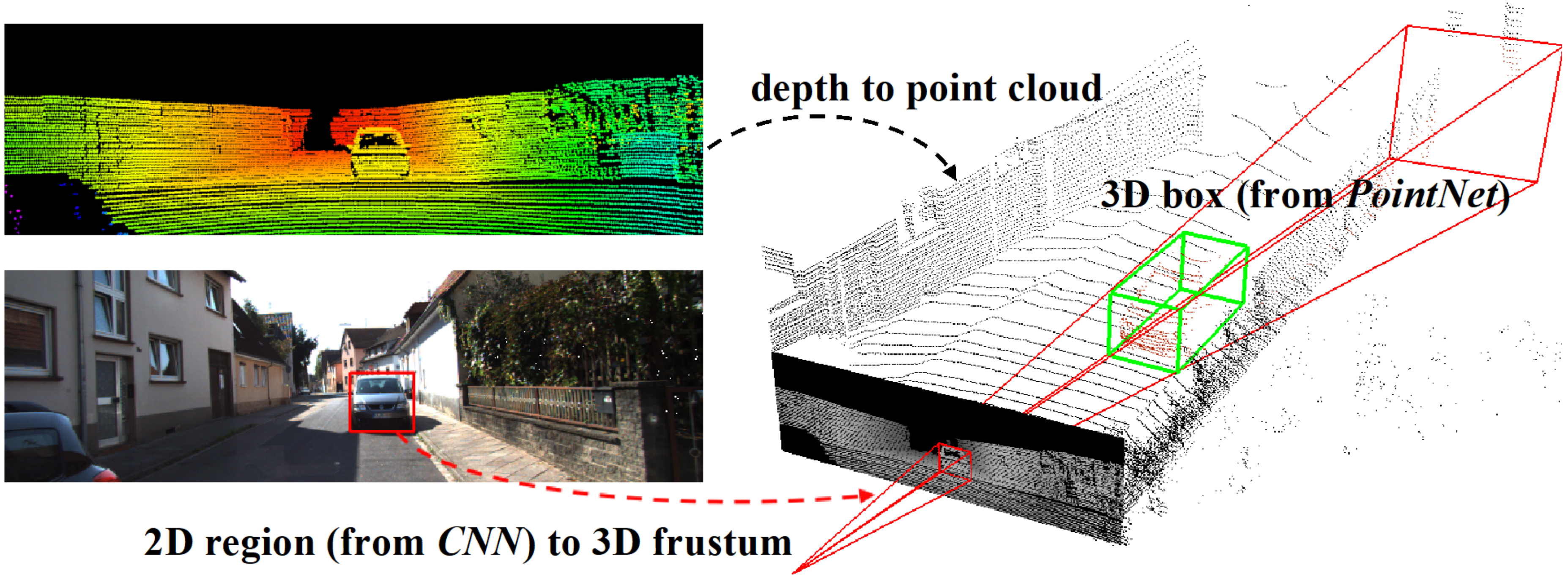


12 Edge
centers



Object Detection in Point Clouds, Outdoors

Frustum PointNets for 3D Object Detection

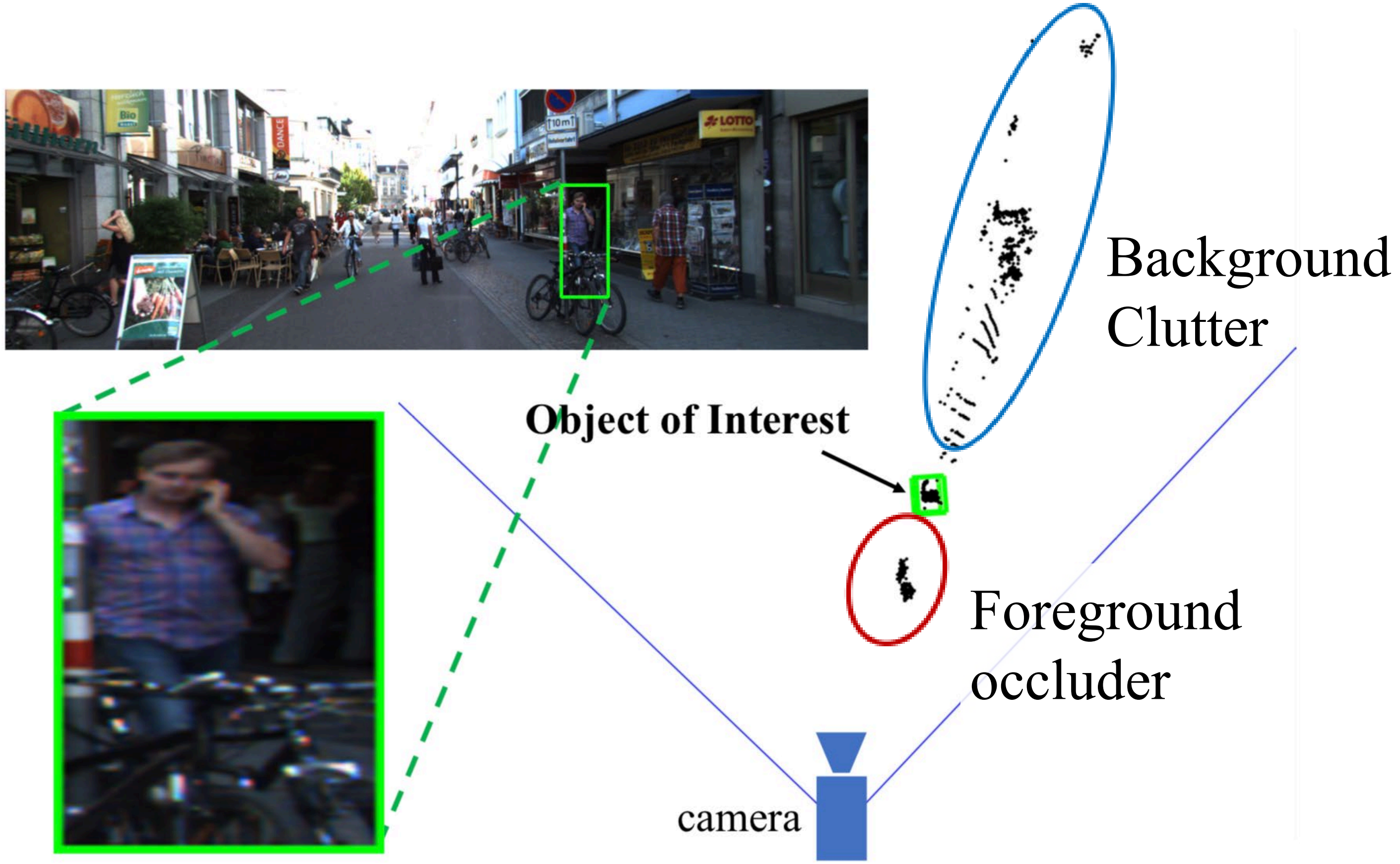


+ Leveraging mature 2D detectors for region proposal. greatly reducing 3D search space.

+ 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)

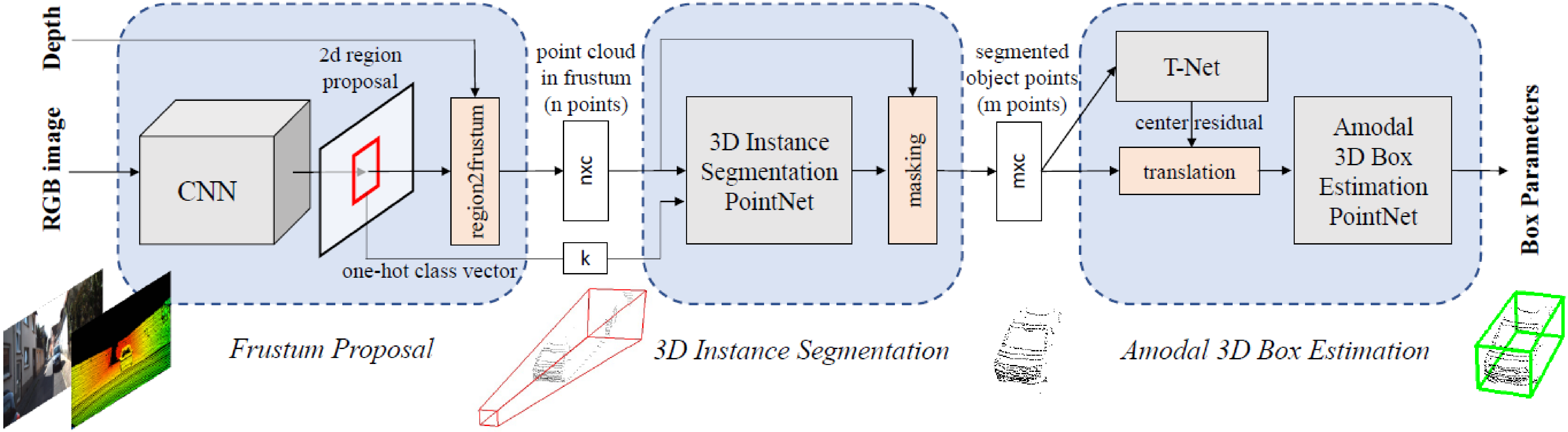
Frustum-based 3D Object Detection: Challenges



- 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

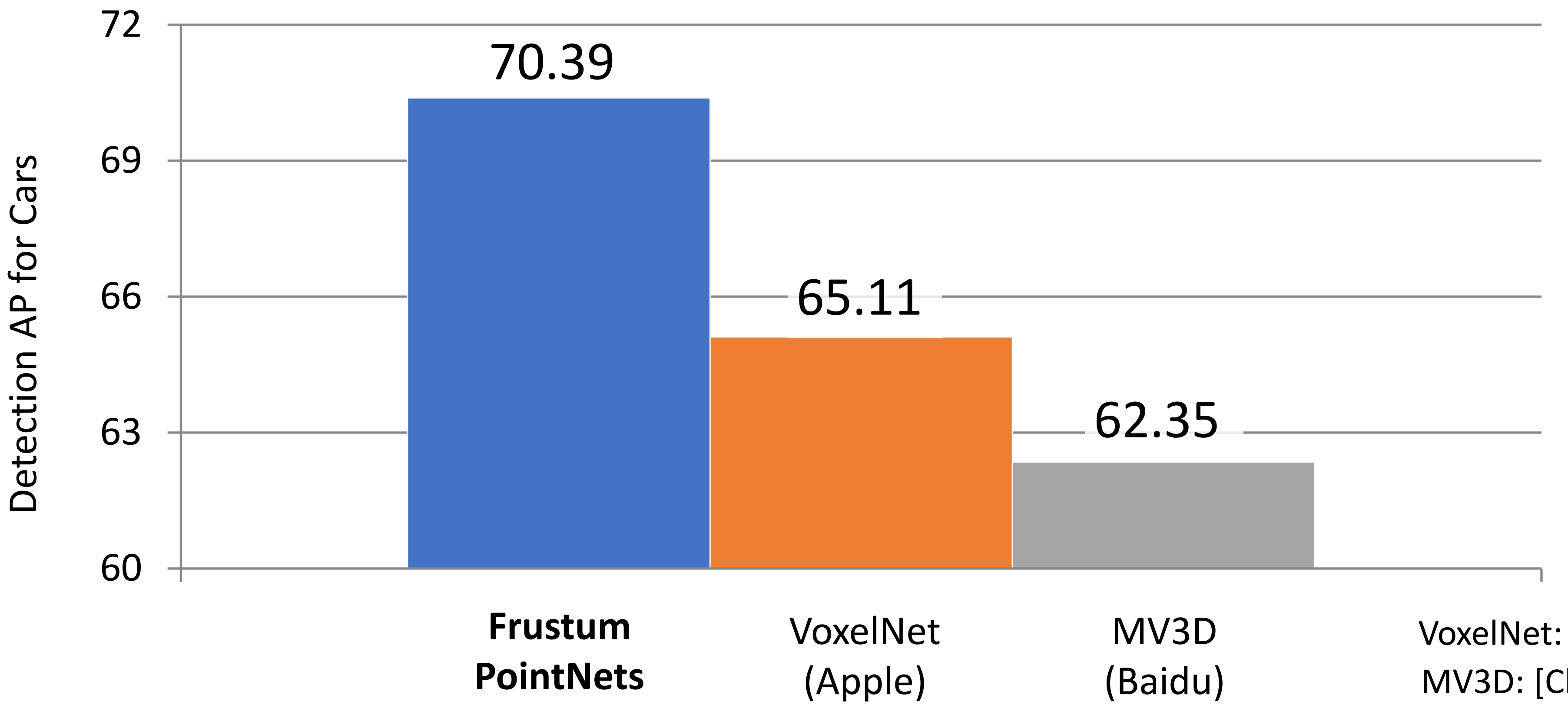
Frustum PointNets: Key to Success

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

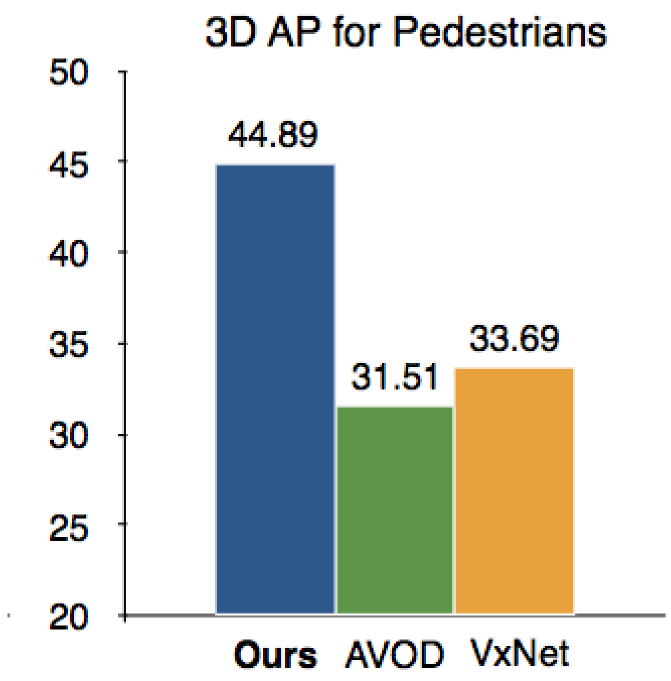


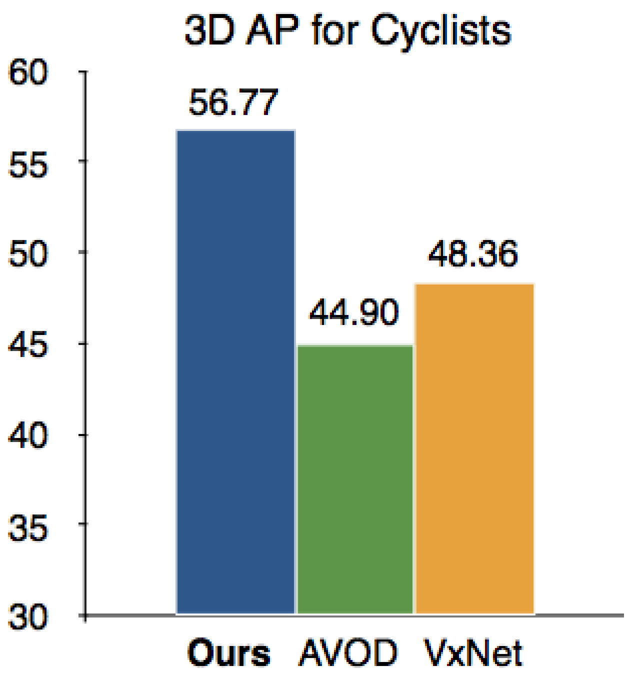
VoxelNet: [Zhou et al. 2018]
MV3D: [Chen et al. 2017]

KITTI Results: Quantitative

Leading performance on KITTI benchmark

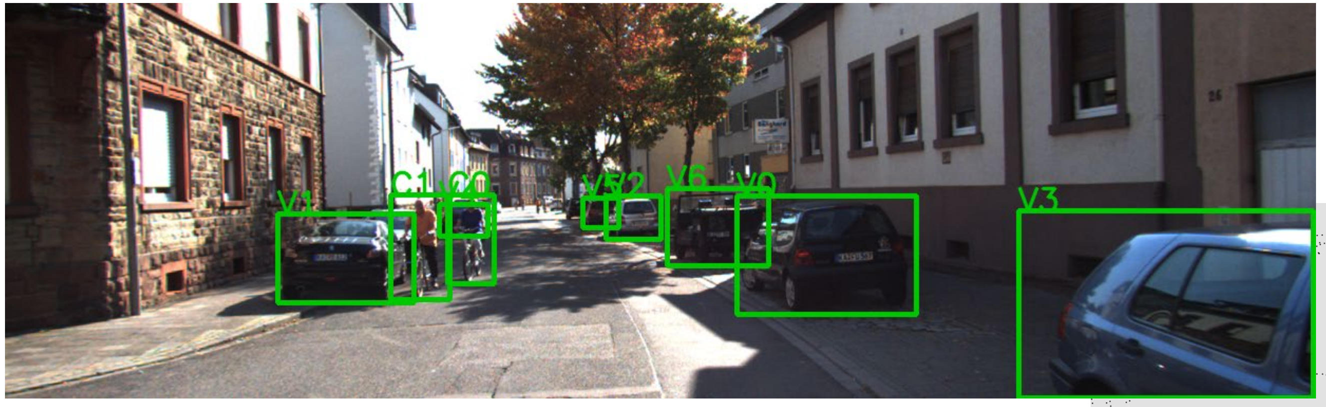
Especially leading at smaller objects (pedestrians and cyclists)
– hard to localize with 3D proposals only.

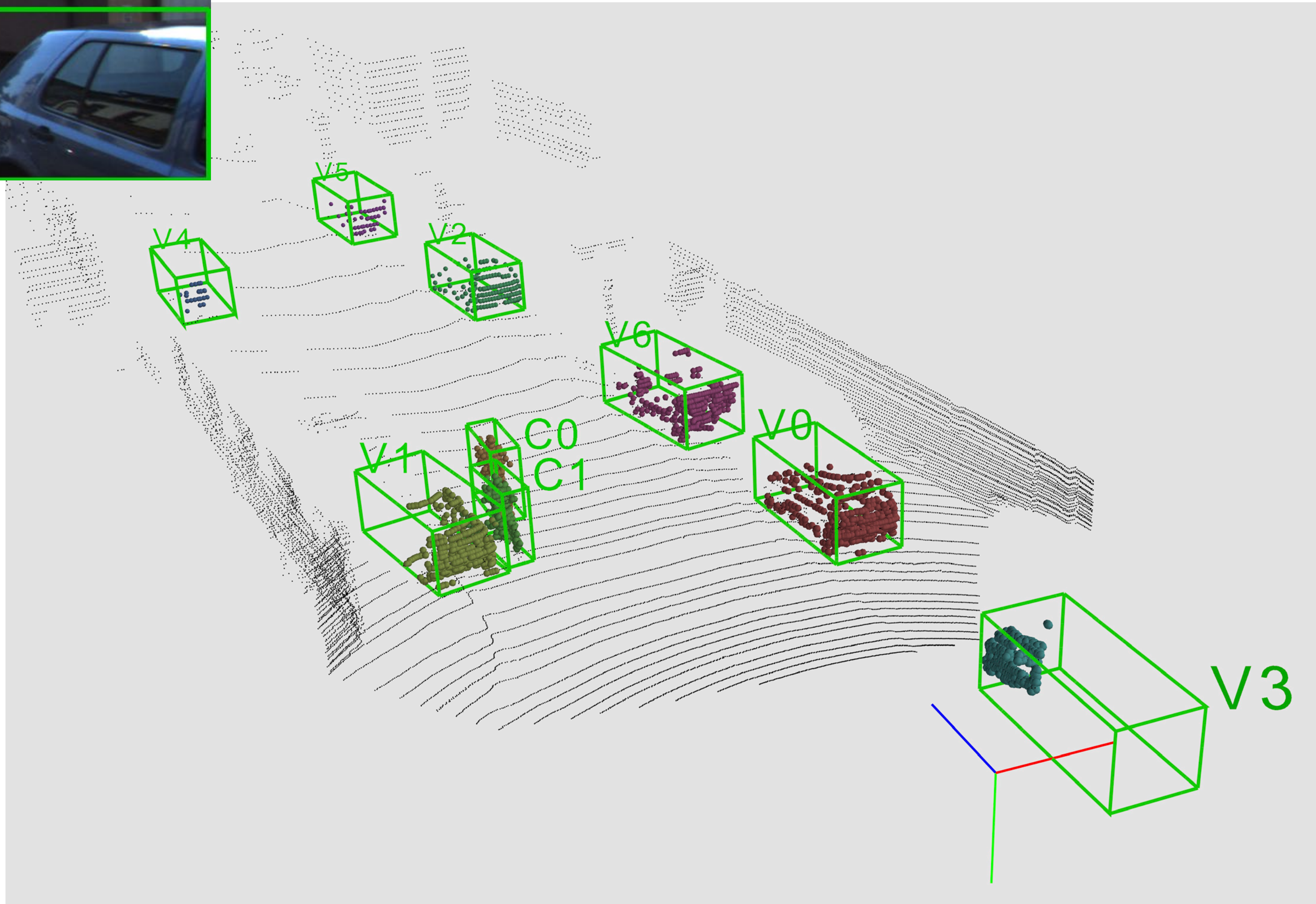




AVOD: [Ku et al. 2018]
VxNet: [Zhou et al. 2017]

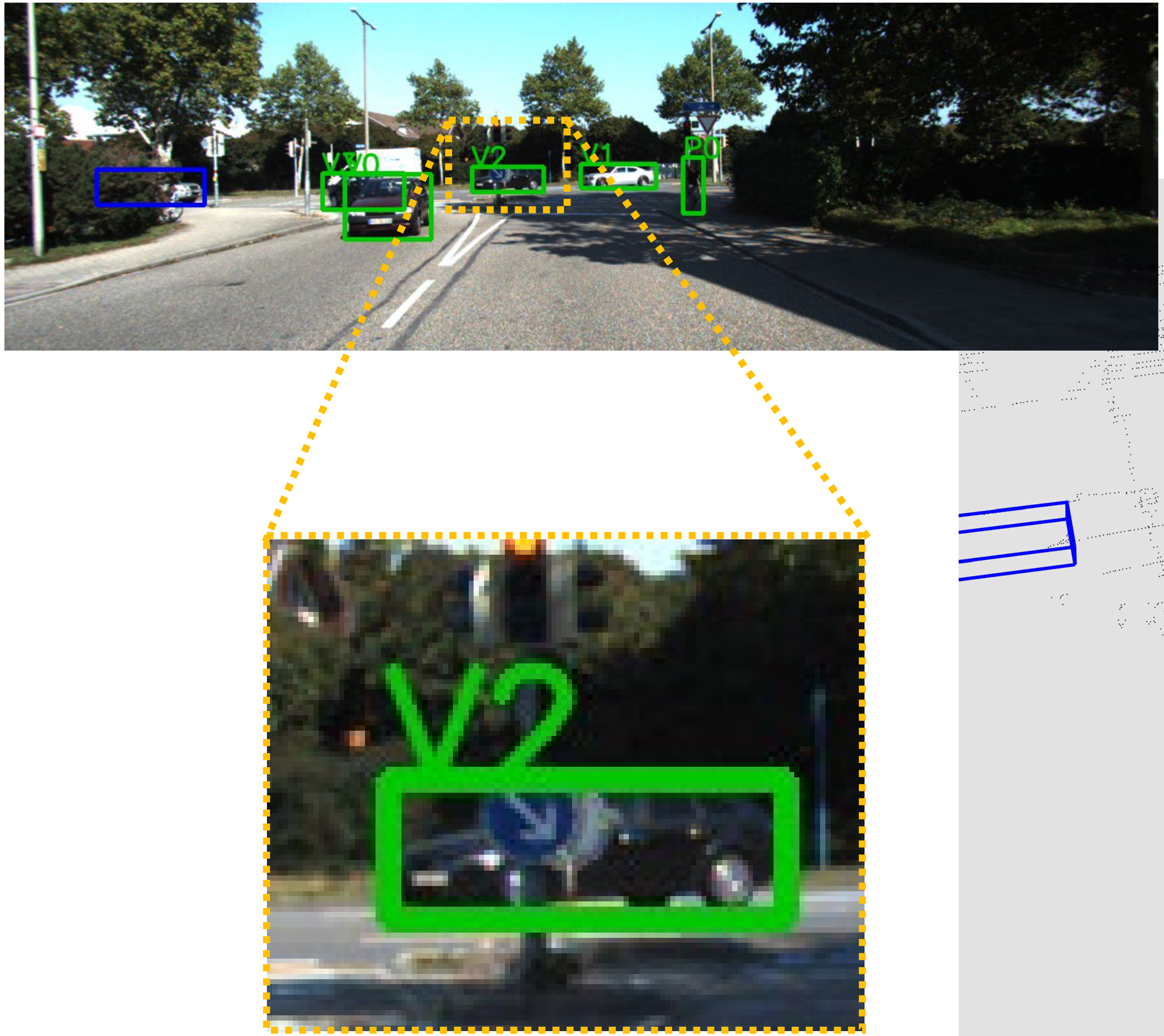
KITTI Results: Qualitative



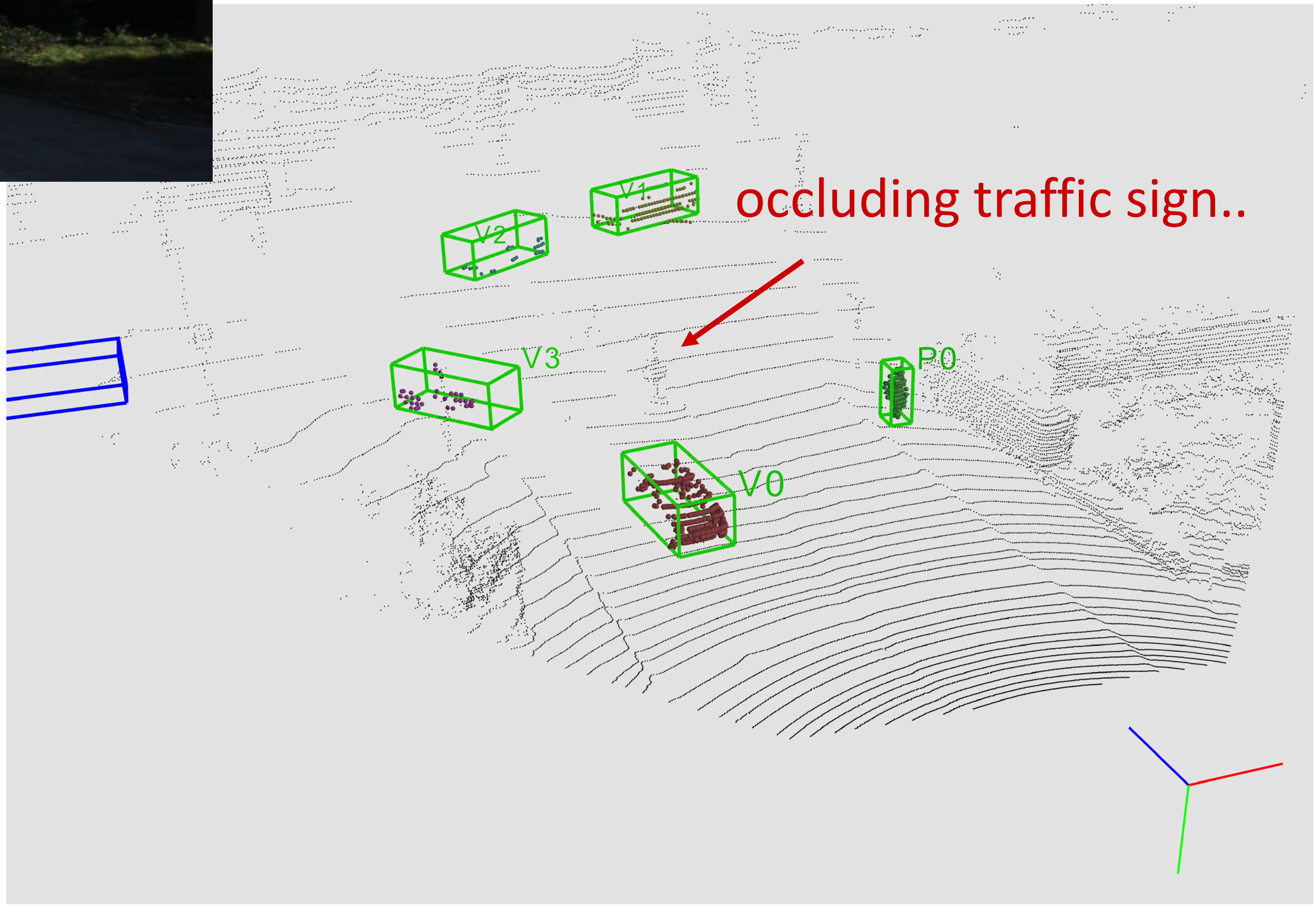


Remarkable box estimation accuracy even with a dozen of points or with very partial point clouds.

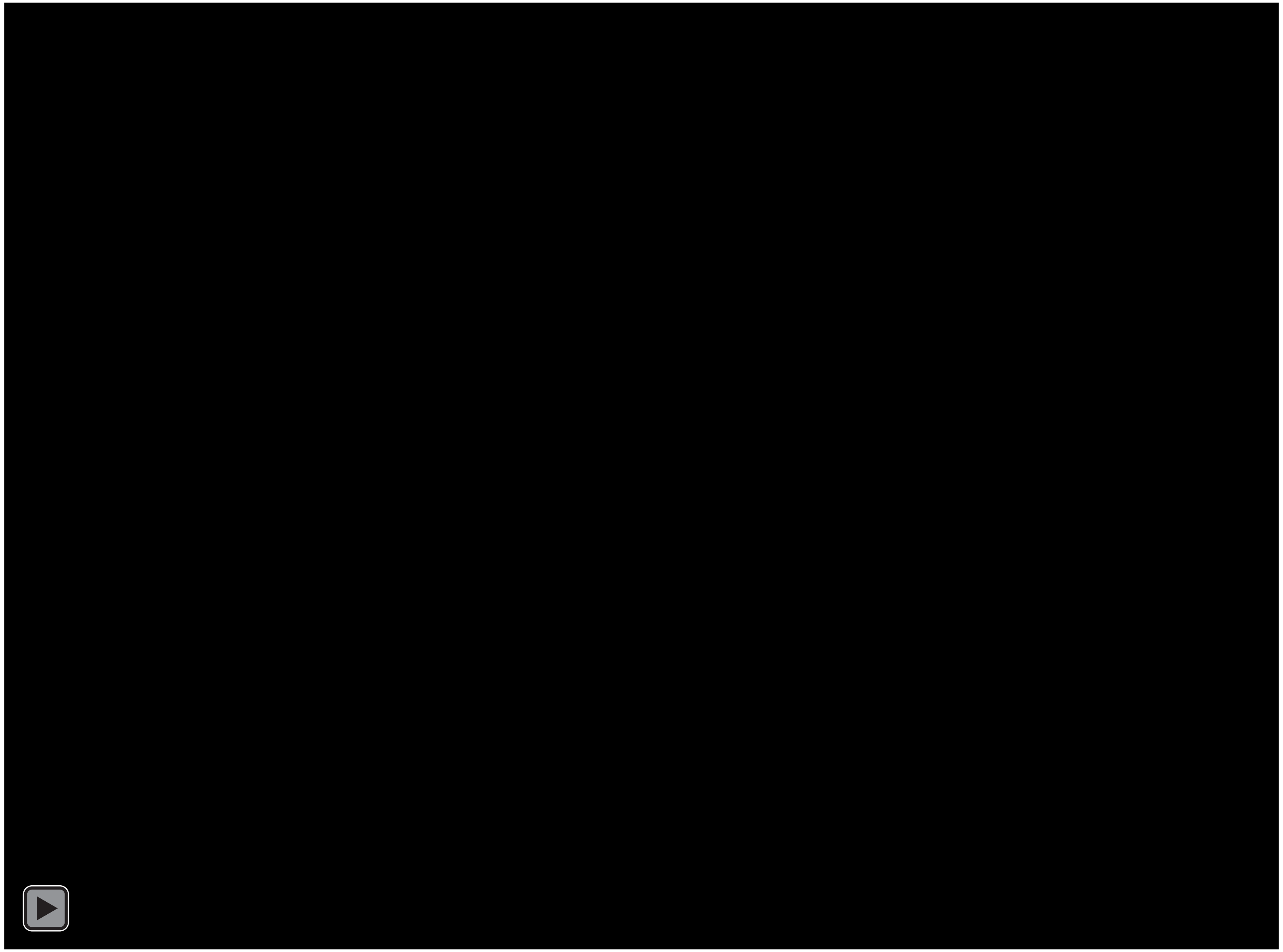
KITTI Results: Qualitative



Correct segmentation in point clouds with heavy occlusion.



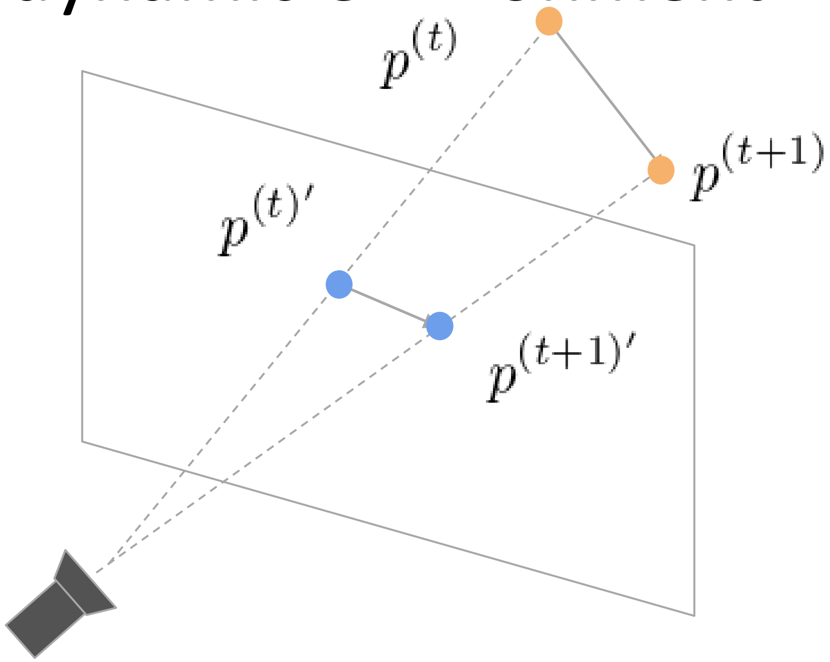
KITTI Results: Example



3D Motion in Point Clouds

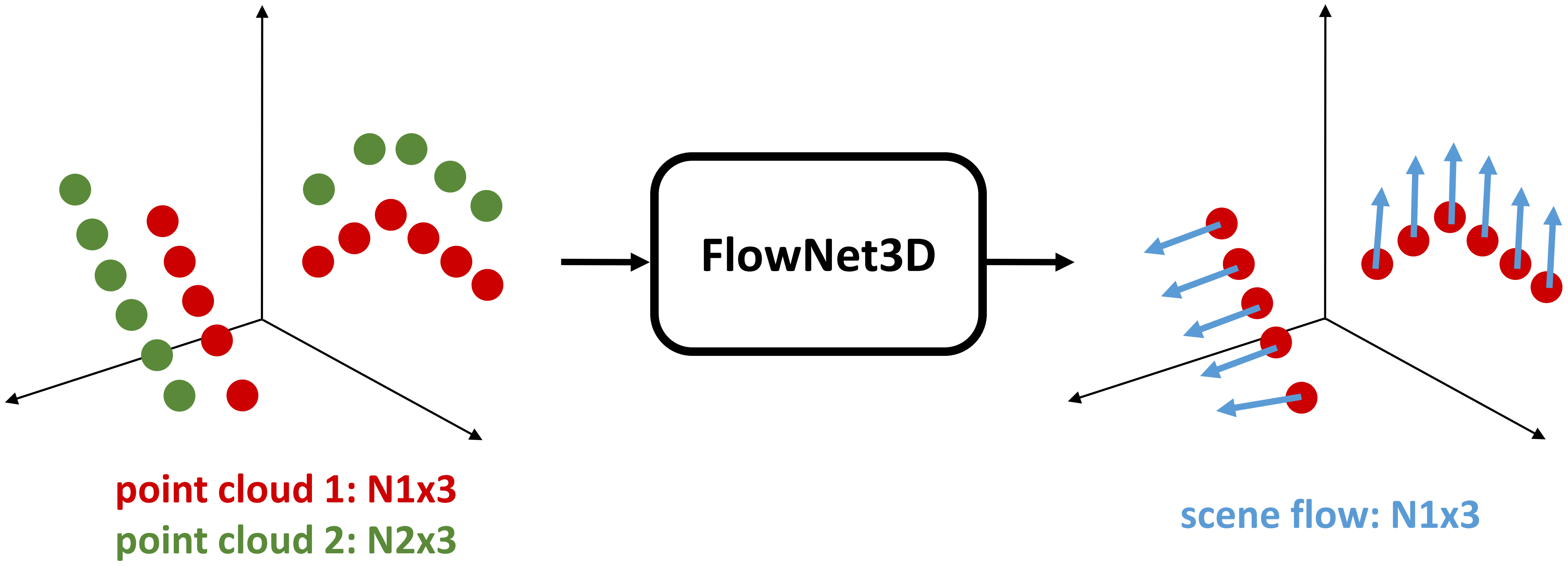
Scene Flow [Vedula et al. 1999]

- 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.



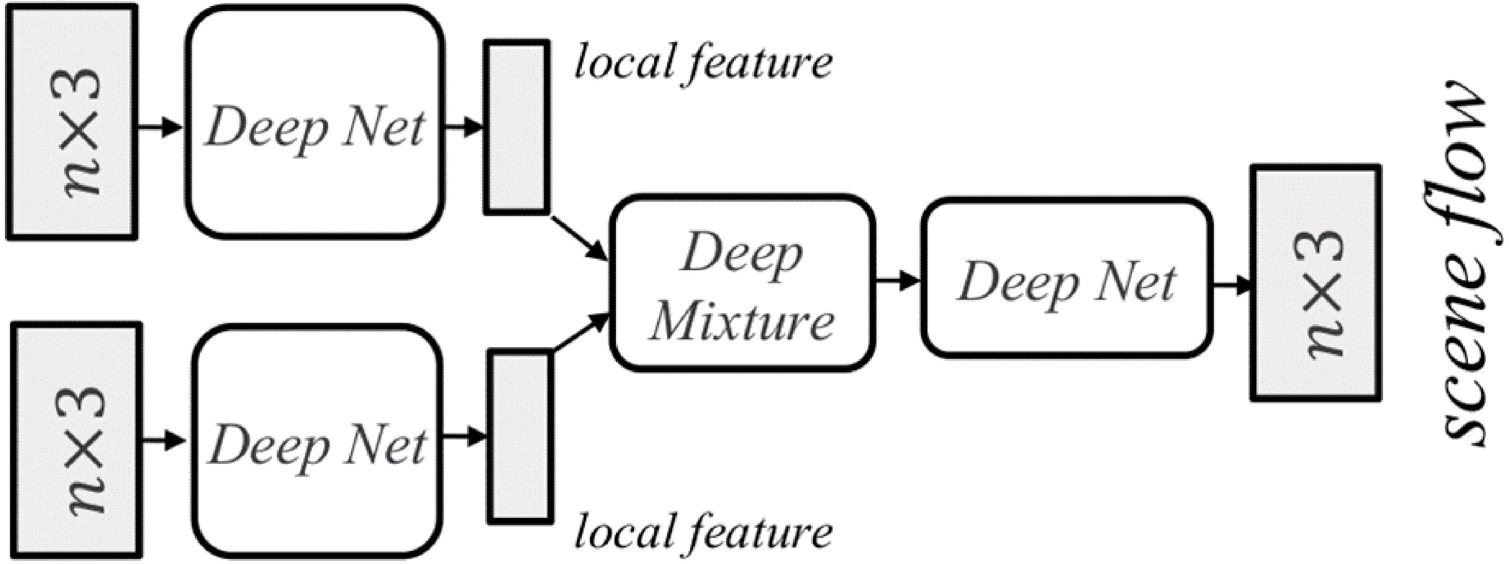
Xingyu Liu, Charles R. Qi, Leonidas Guibas. Learning Scene Flow in 3D Point Clouds, (CVPR 2019).

Deep Net Architecture

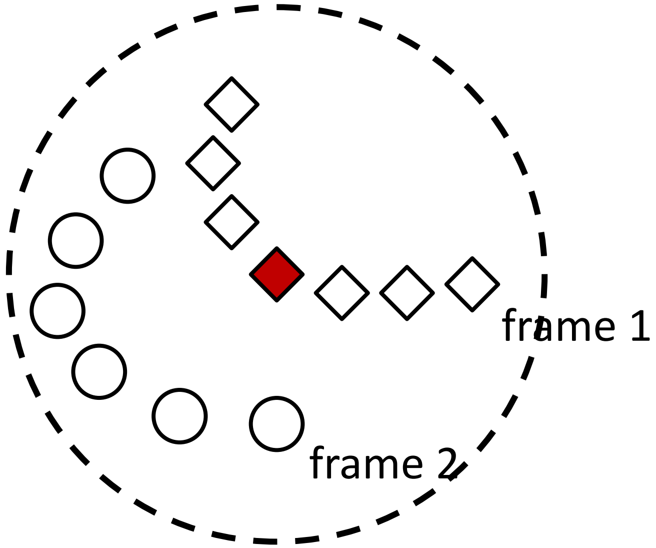
- How to learn point cloud features?
- Where in the network architecture to mix point features from consecutive frames?
- How to mix them?

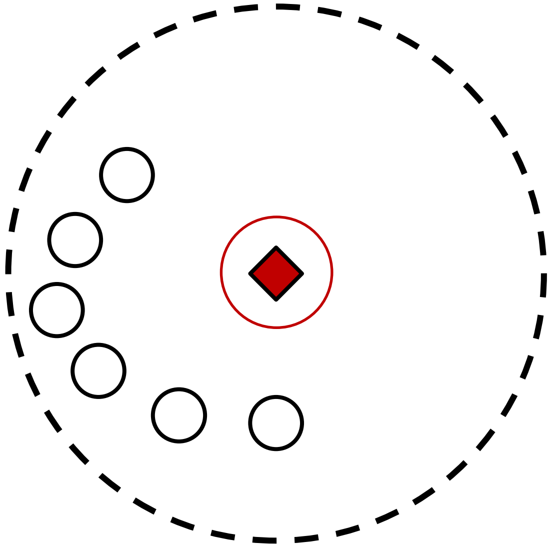
Intermediate level

Middle-level
deep mixture

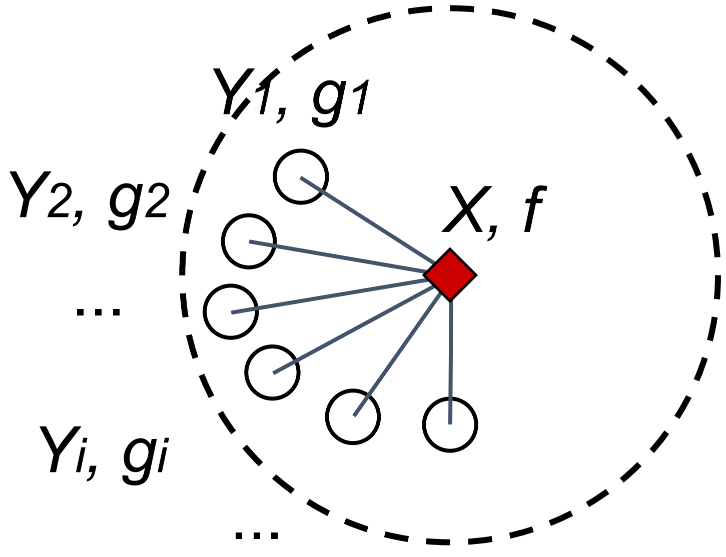


Middle-Level Mixing





Point Attributes



dist(g_1, f), Y_1-X

dist(g_2, f), Y_2-X

⋮

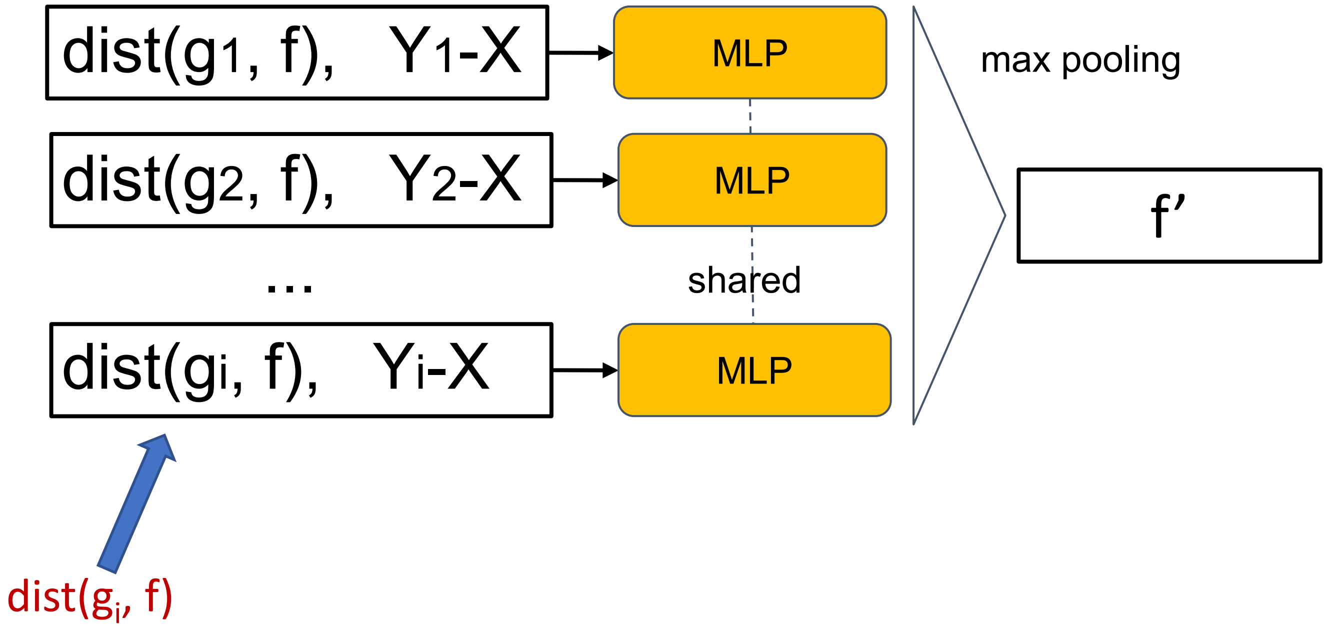
dist(g_i, f), Y_i-X

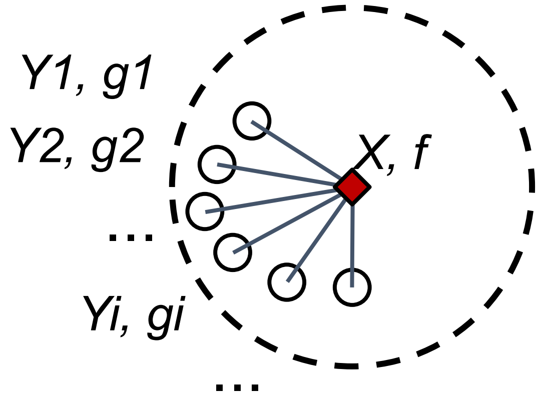
⋮

Naive approach: concatenation

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

A More Structured Approach



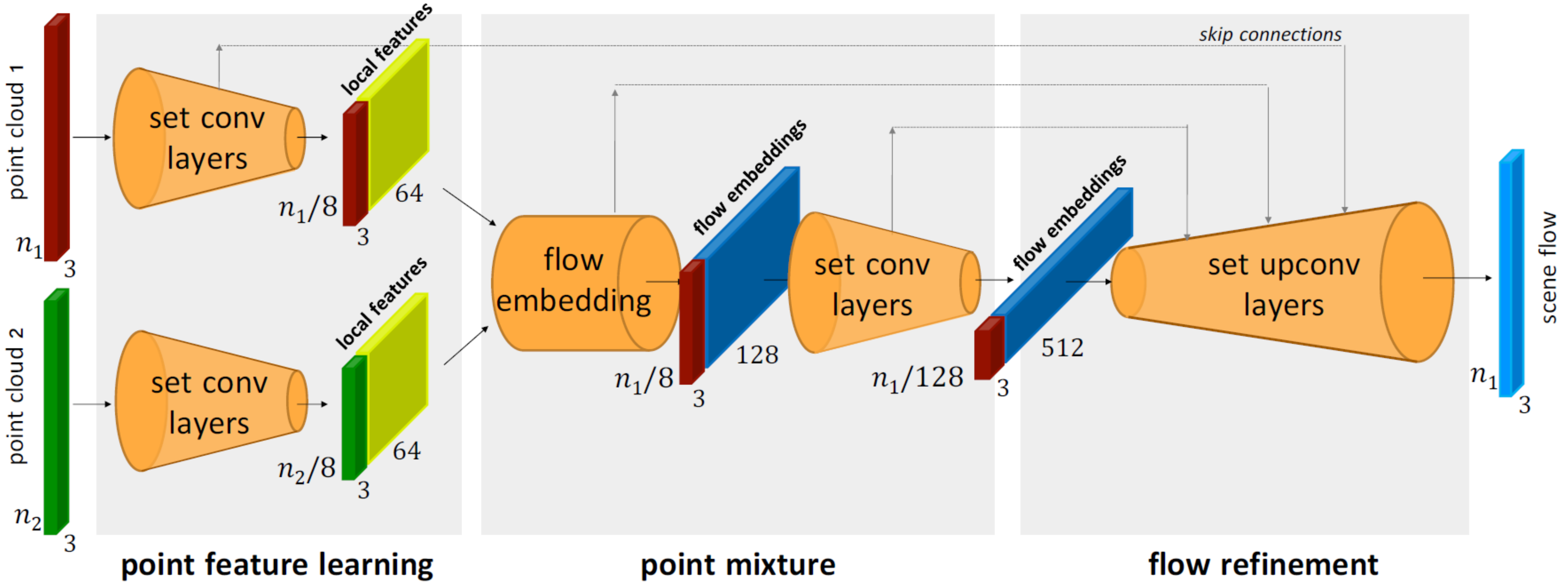


"Distance" functions:

- 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 …

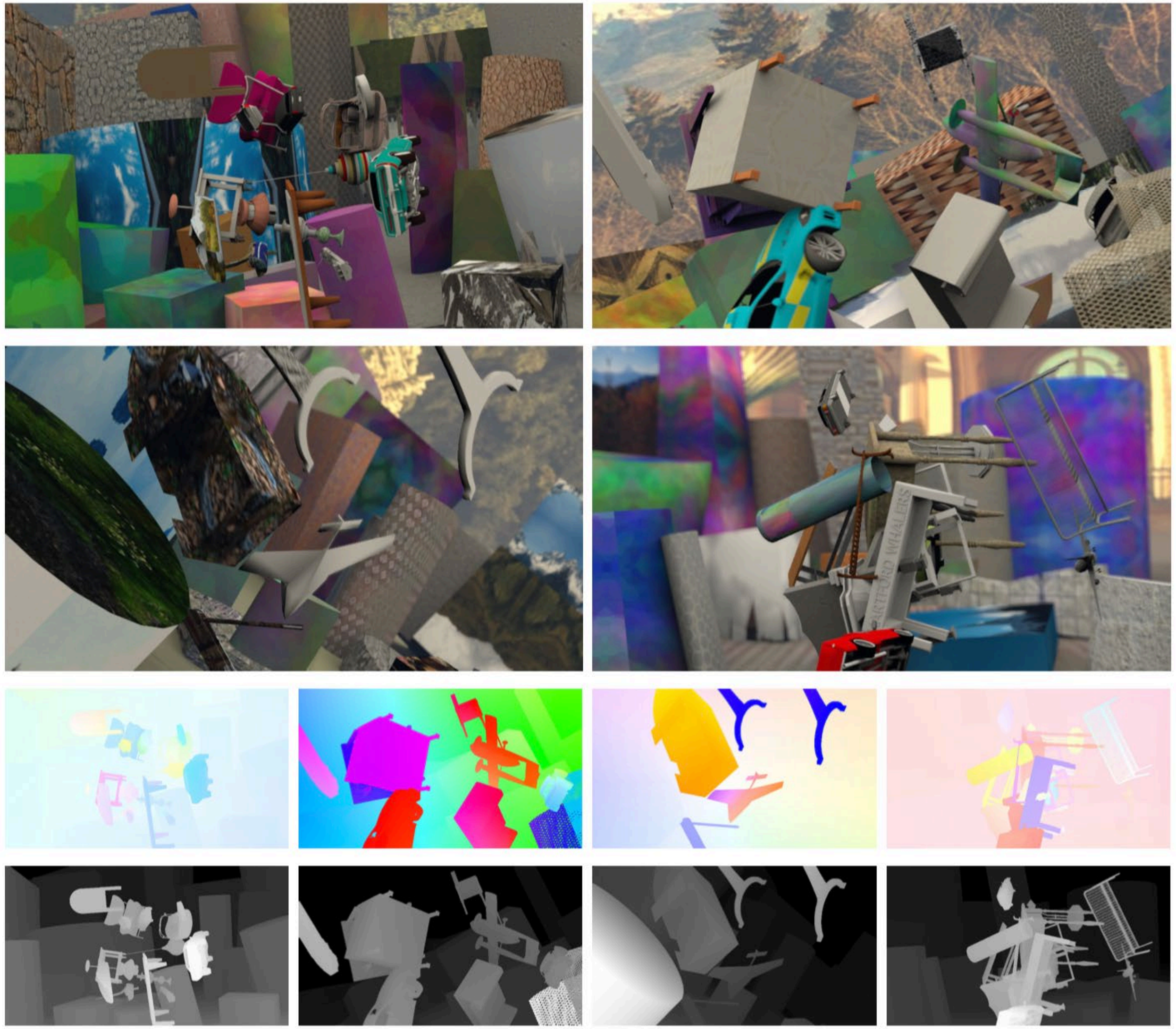
Pointnet++

Training on Synthetic Data

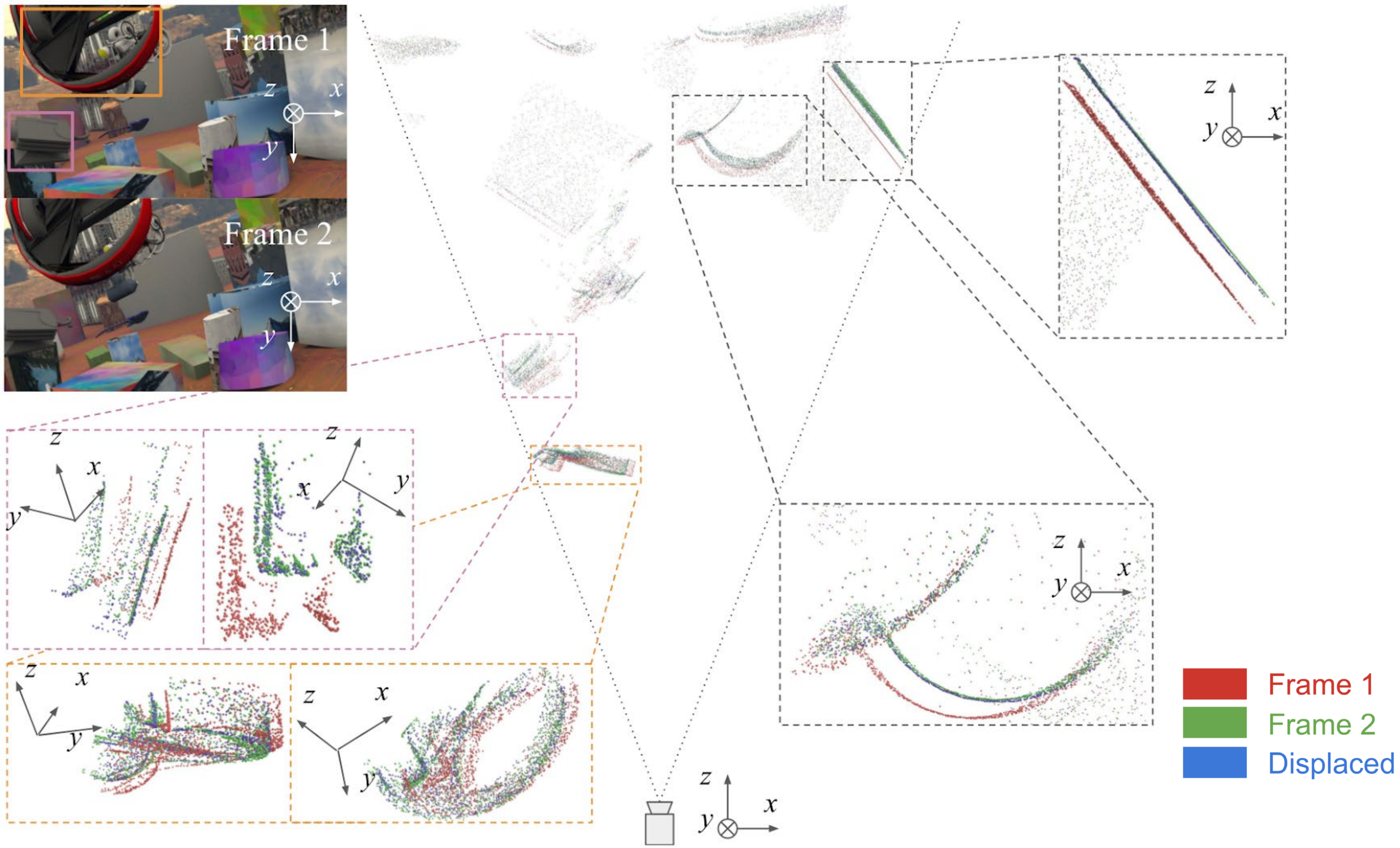
FlyingThings3D [Mayer et al. 2016] dataset from MPI

Random ShapeNet objects

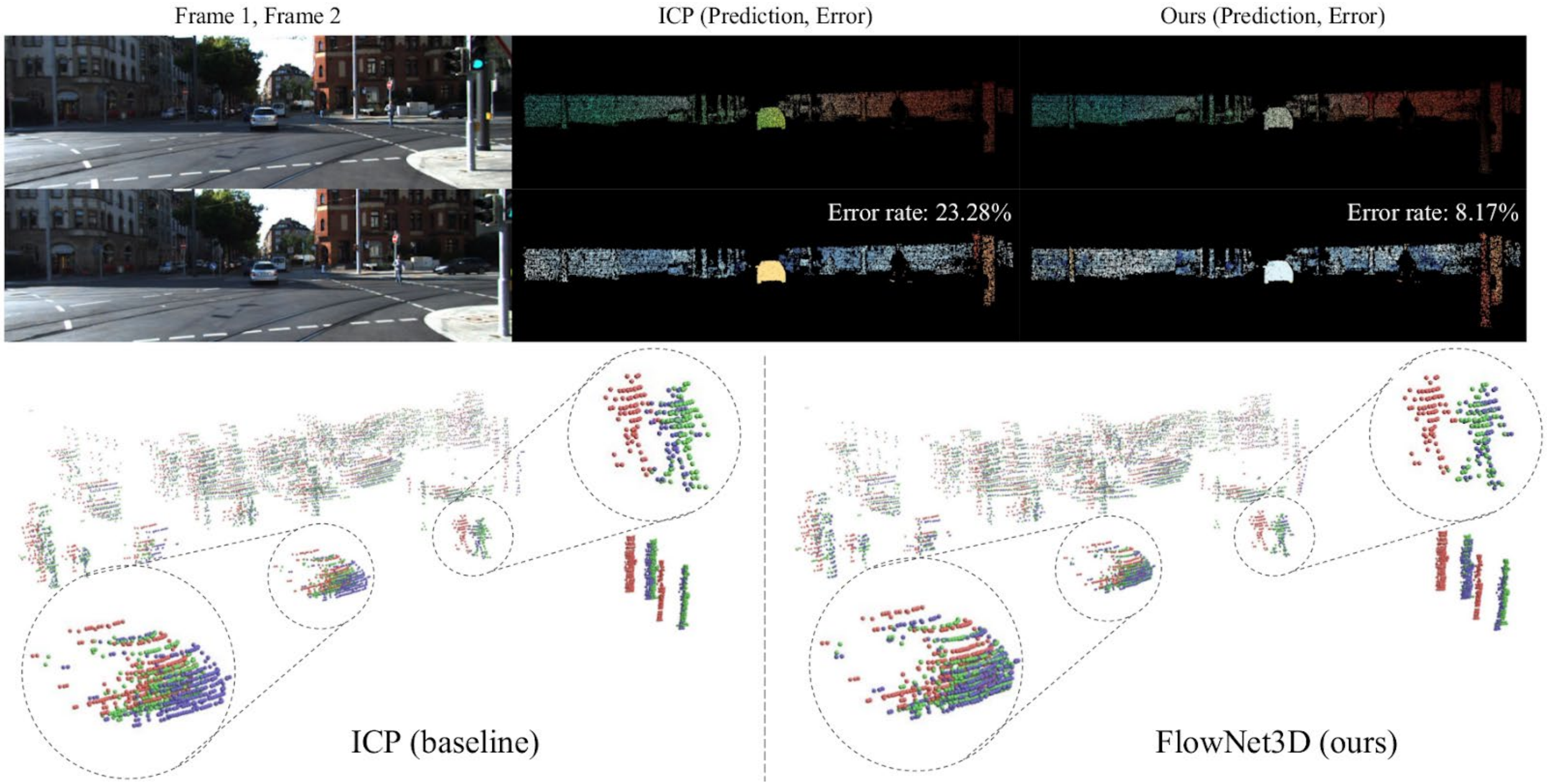
Very challenging dataset with strong occlusions and large motions.



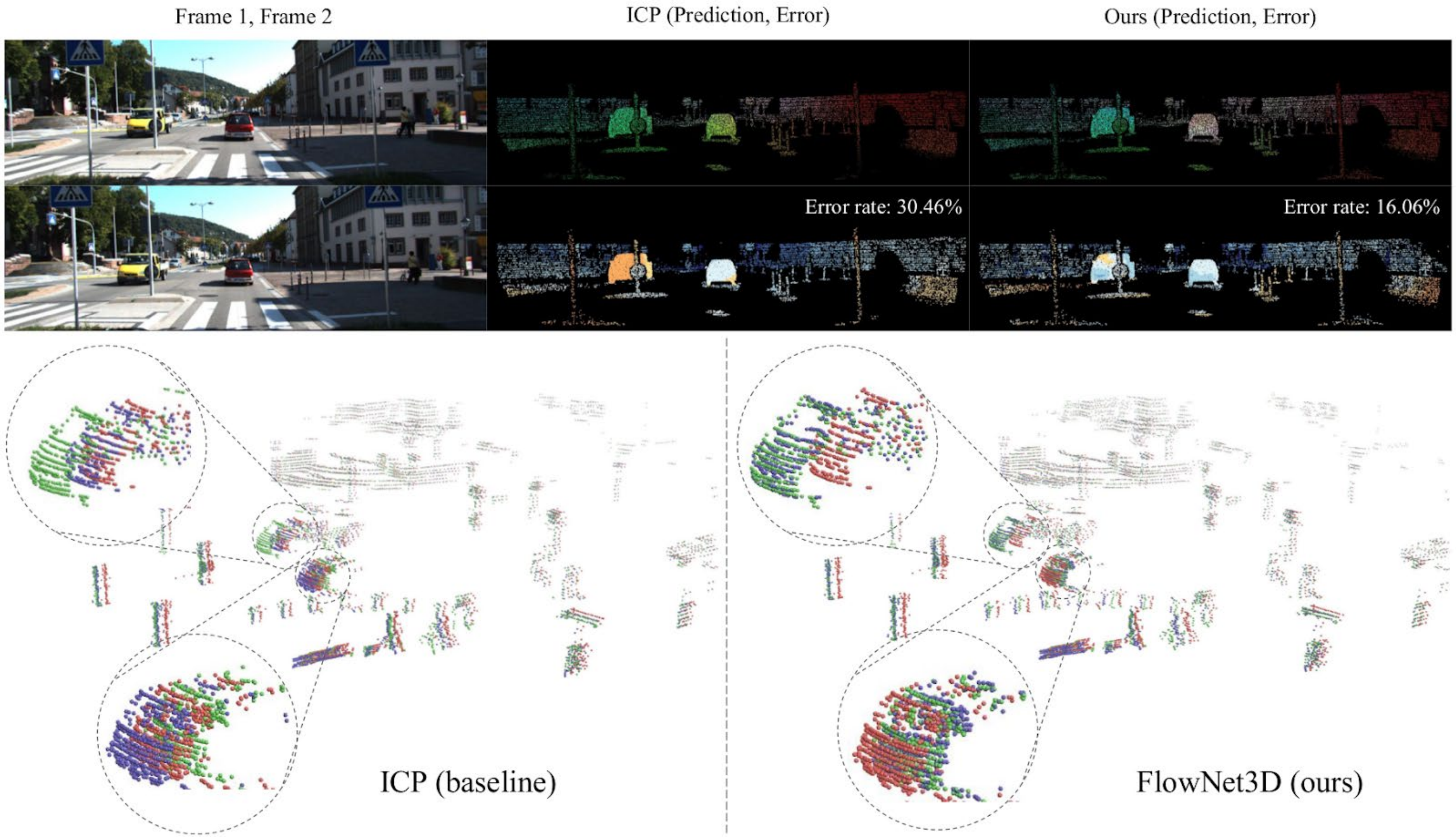
FlyingThings3D Results



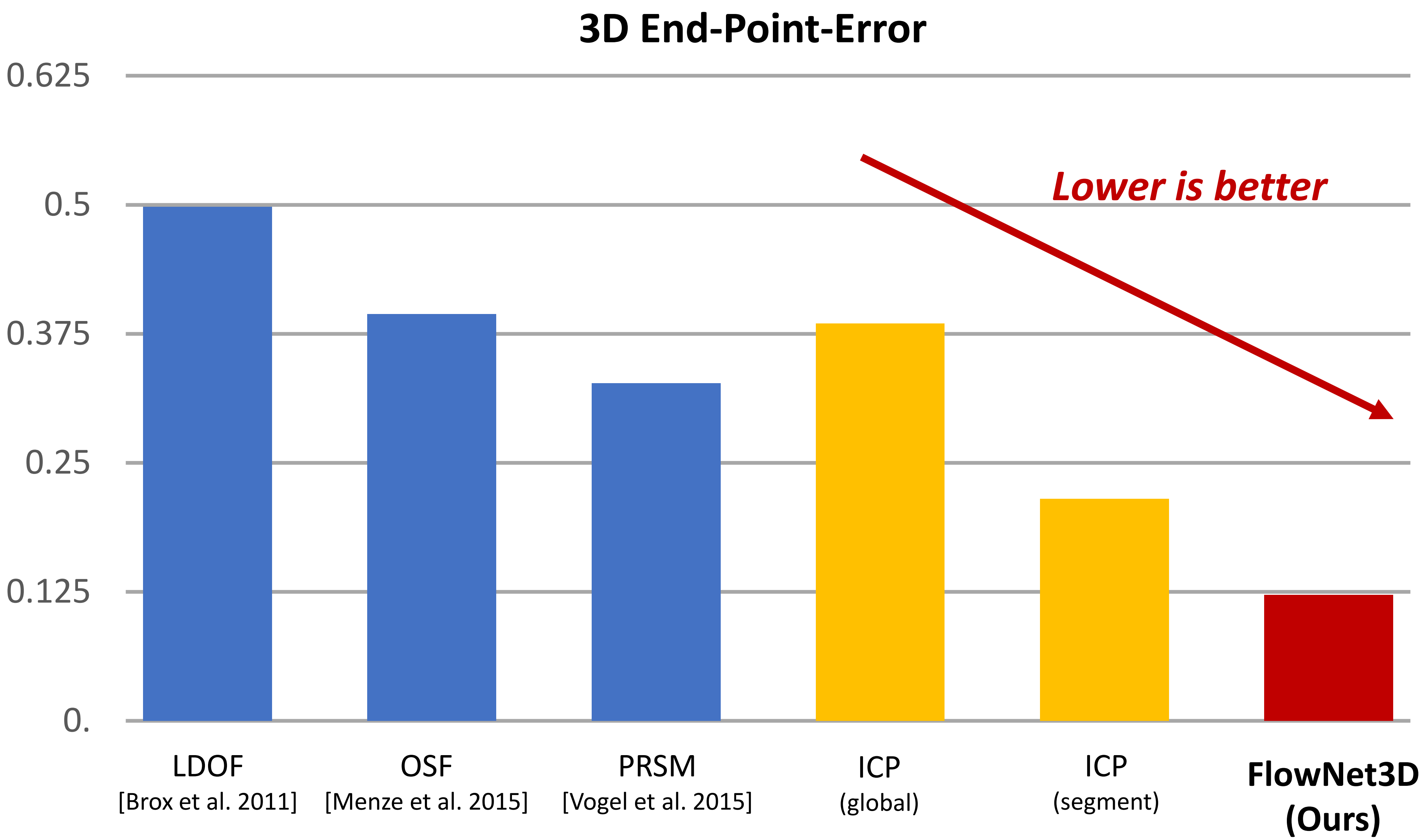
KITTI Results



KITTI Results

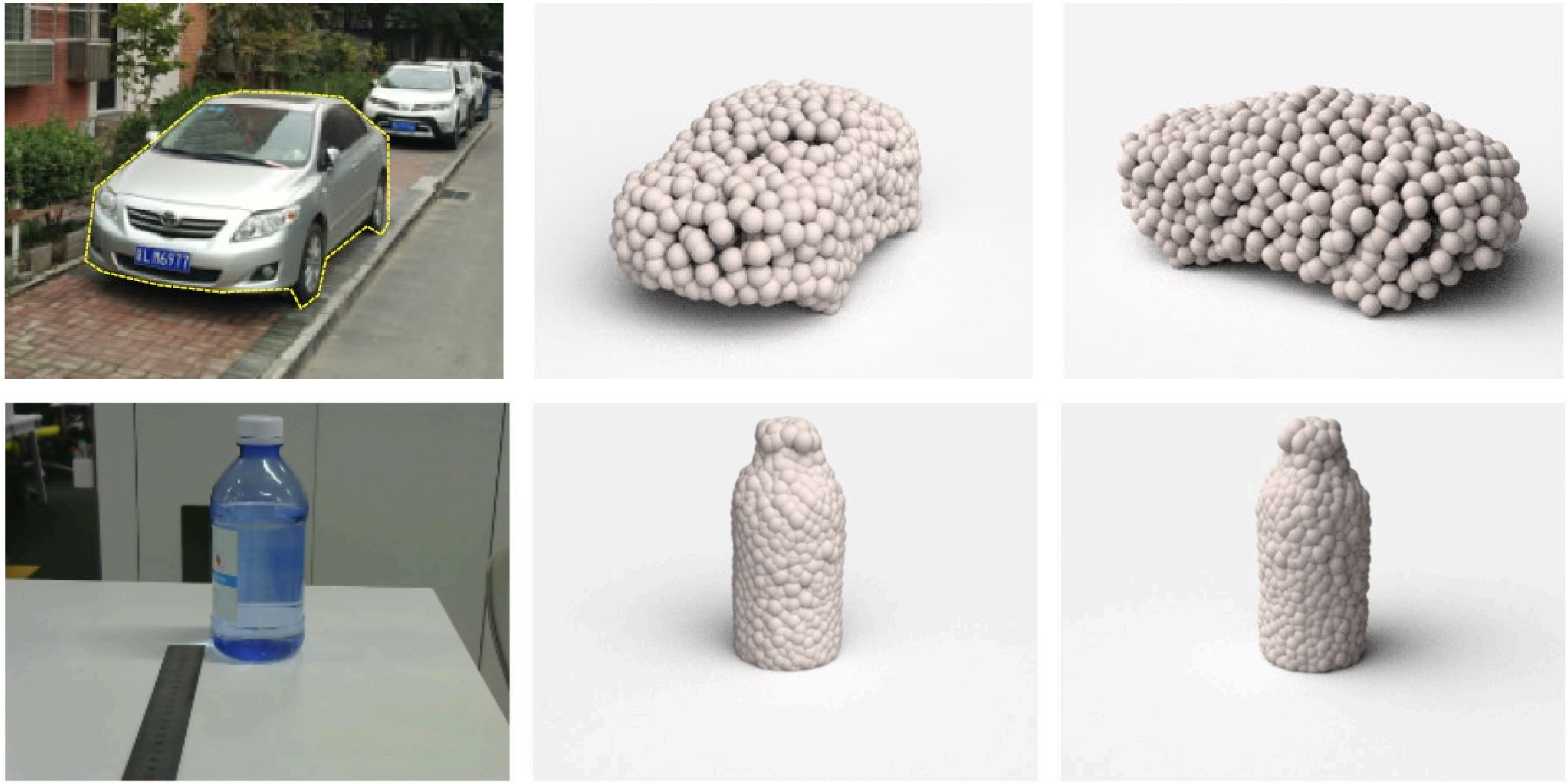


Generalizing to KITTI: Quantitative



Point-Set Generation

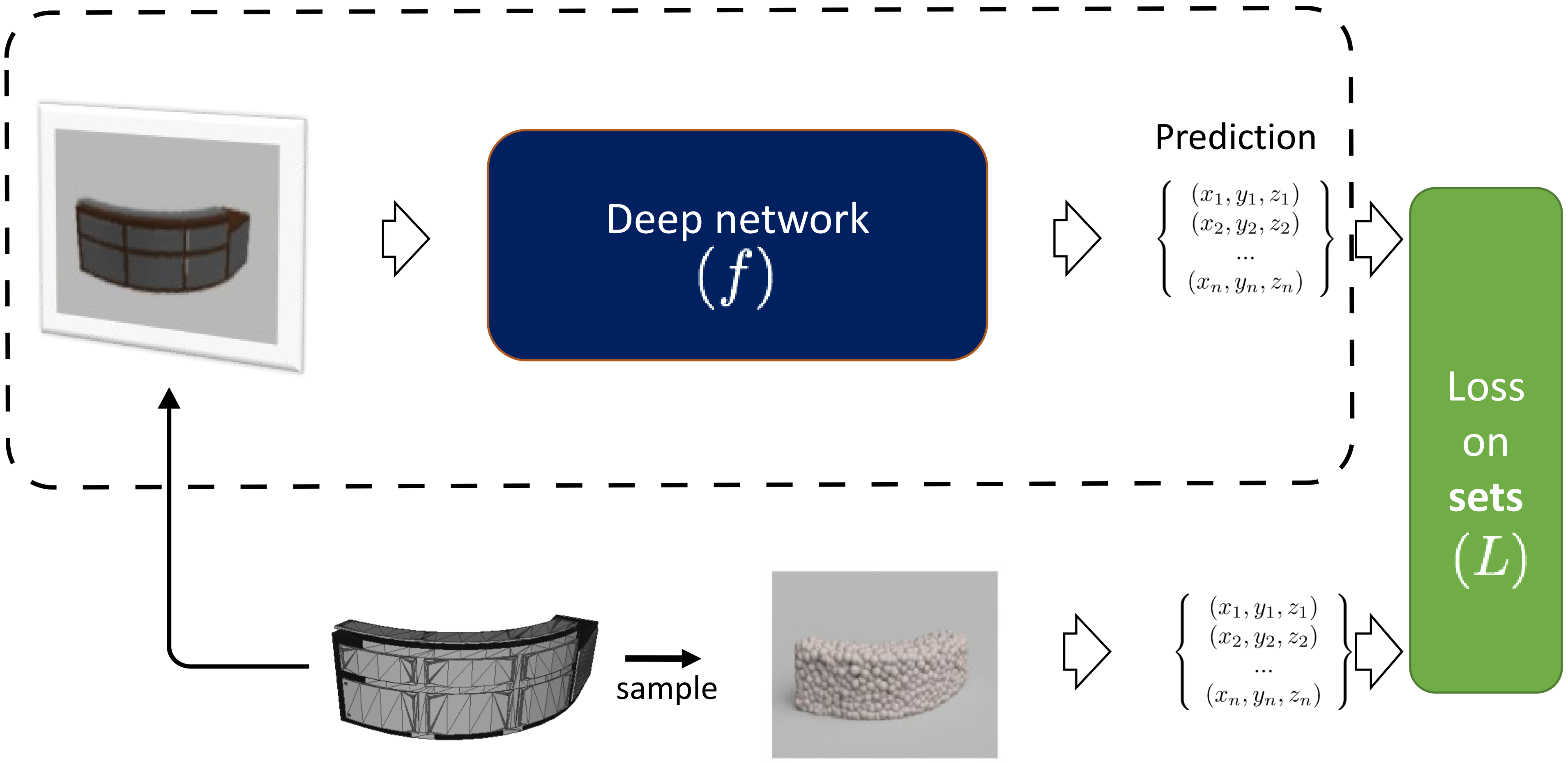
Point Cloud Synthesis from a Single Image



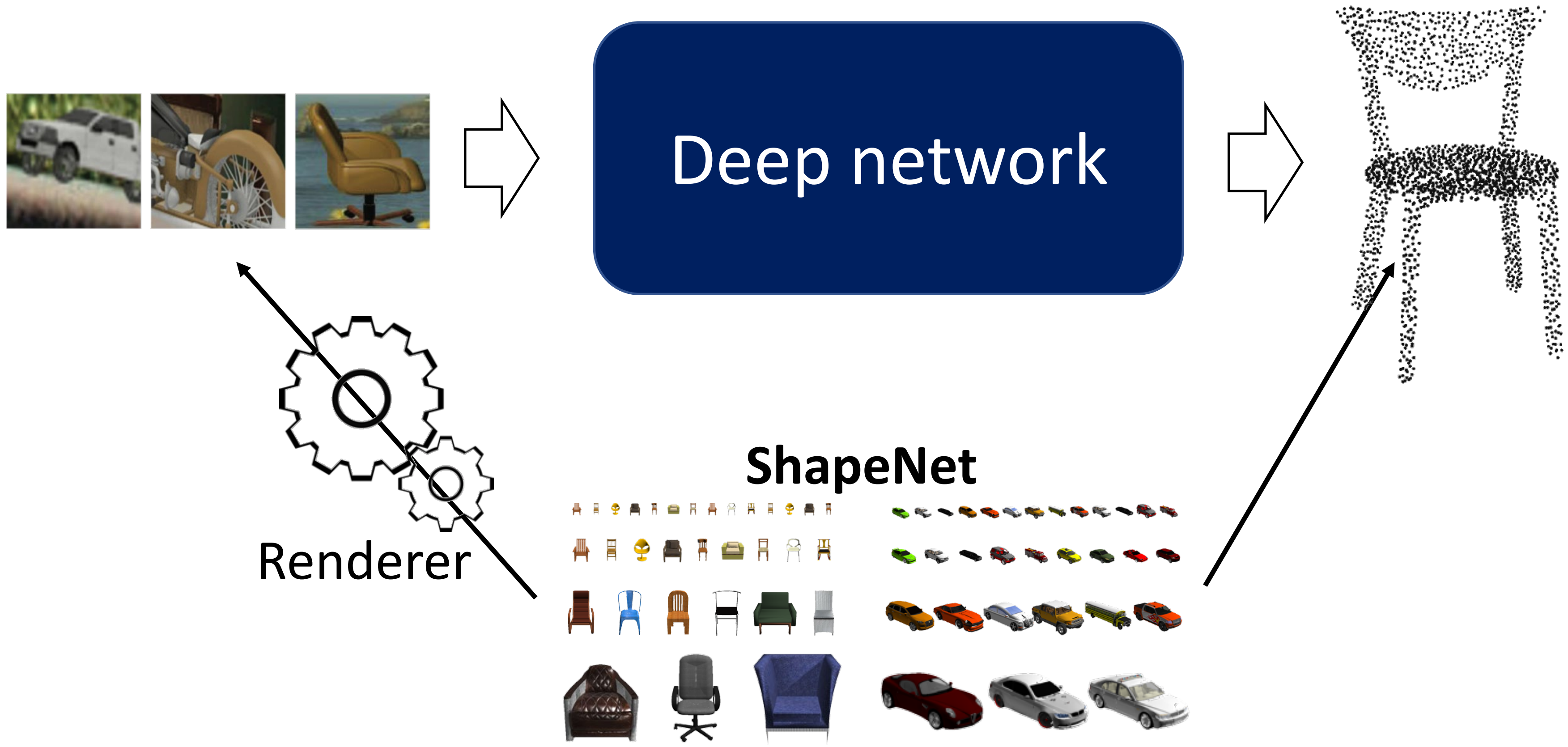
Input Reconstructed 3D point cloud

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

End-to-End Learning

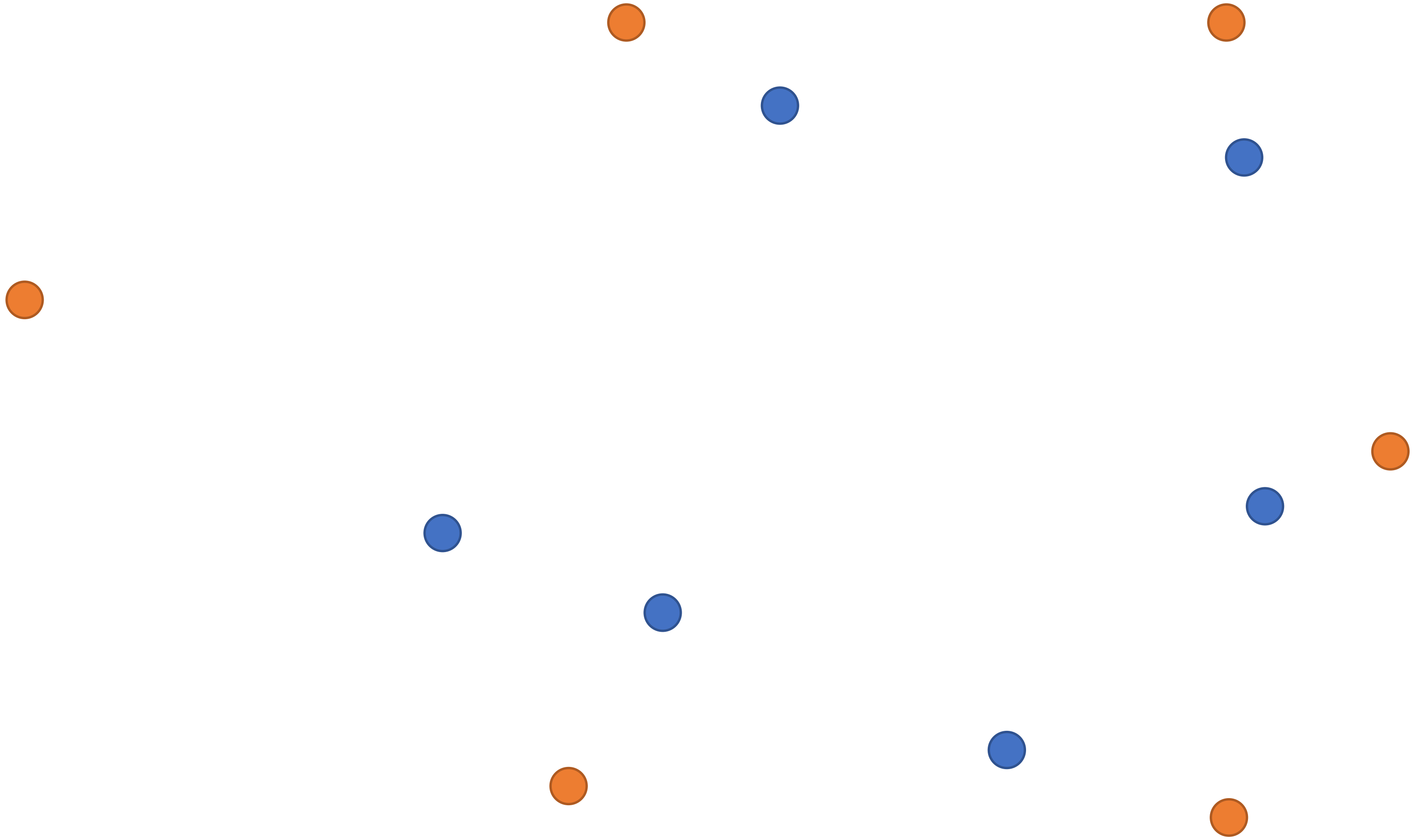


Synthesize for Learning



Distance Metrics Between Point Sets

Given two sets of points, measure their discrepancy

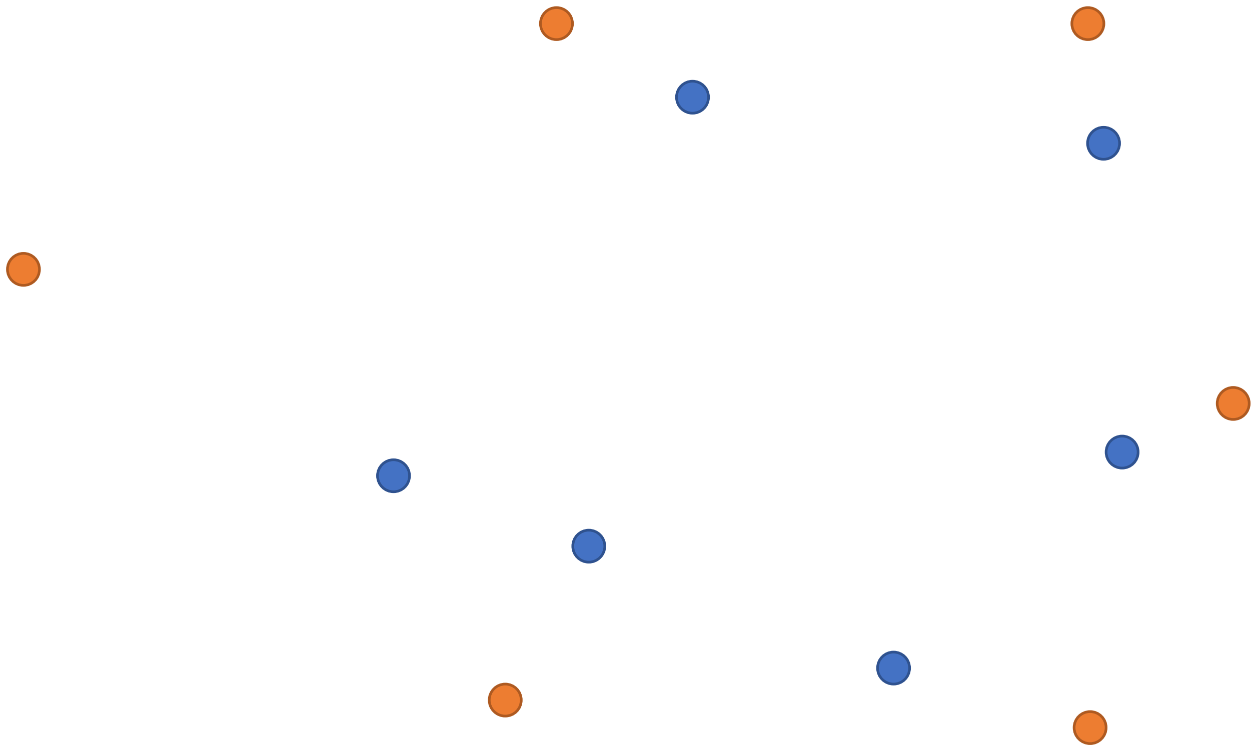


Common Distance Metrics

Worst case: Hausdorff distance (HD)

Average case: Chamfer distance (CD)

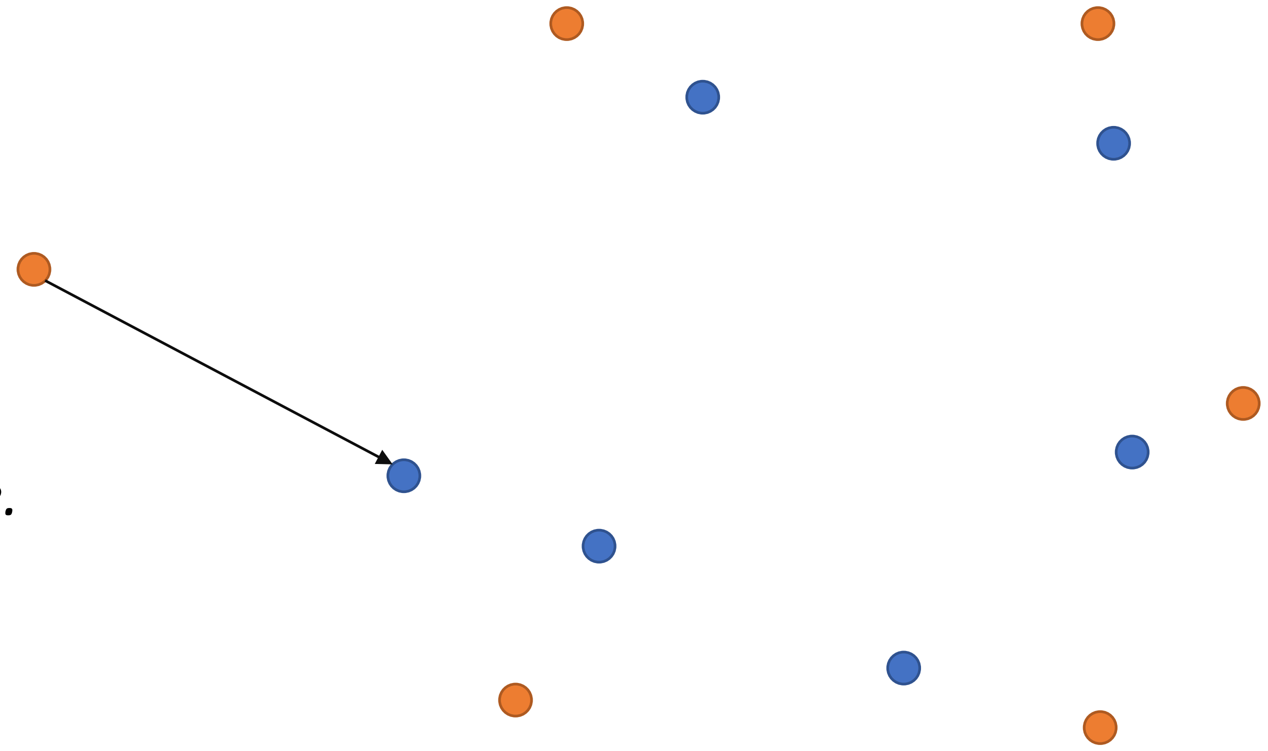
Optimal case: Earth Mover's distance (EMD)



Common Distance Metrics

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

Worst case: Hausdorff distance (HD)



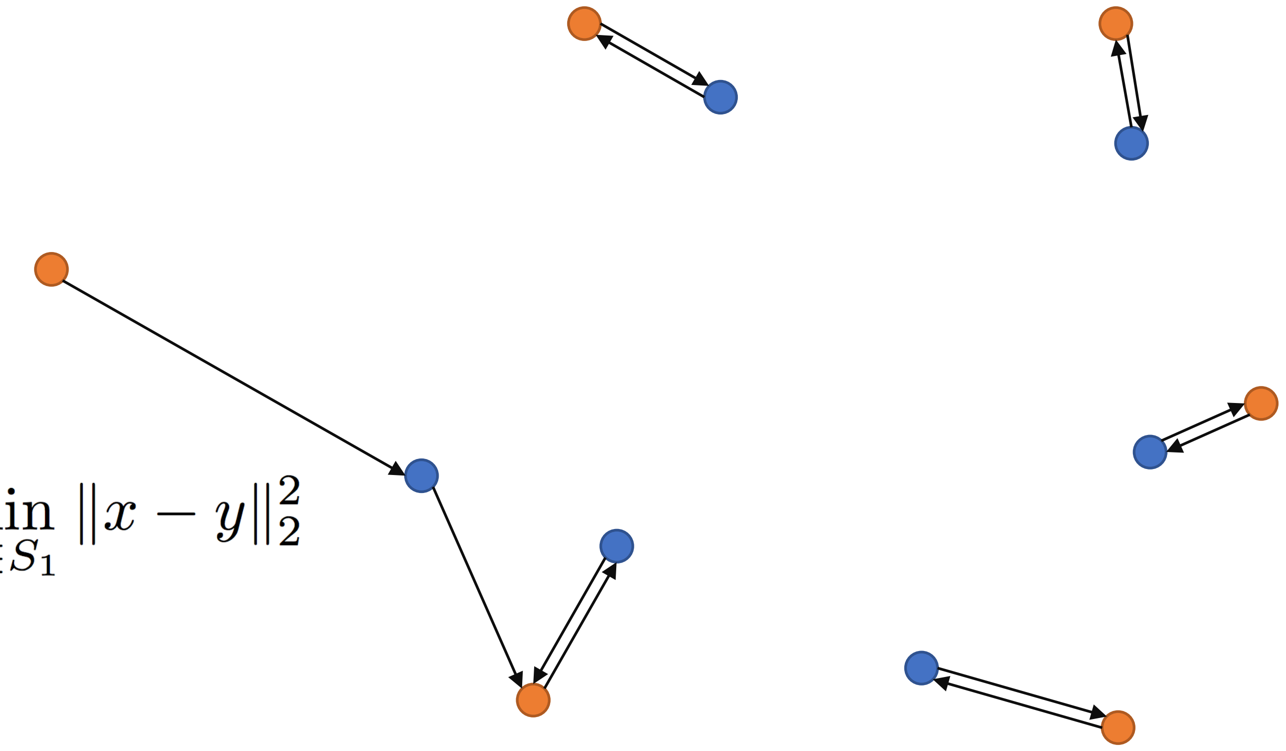
A single distant pair determines the distance.
In other words, ***not robust to outliers!***

Common Distance Metrics

Worst case: Hausdorff distance (HD)

Average case: Chamfer distance (CD)

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



Average all the nearest neighbor distance by nearest neighbors

Common Distance Metrics

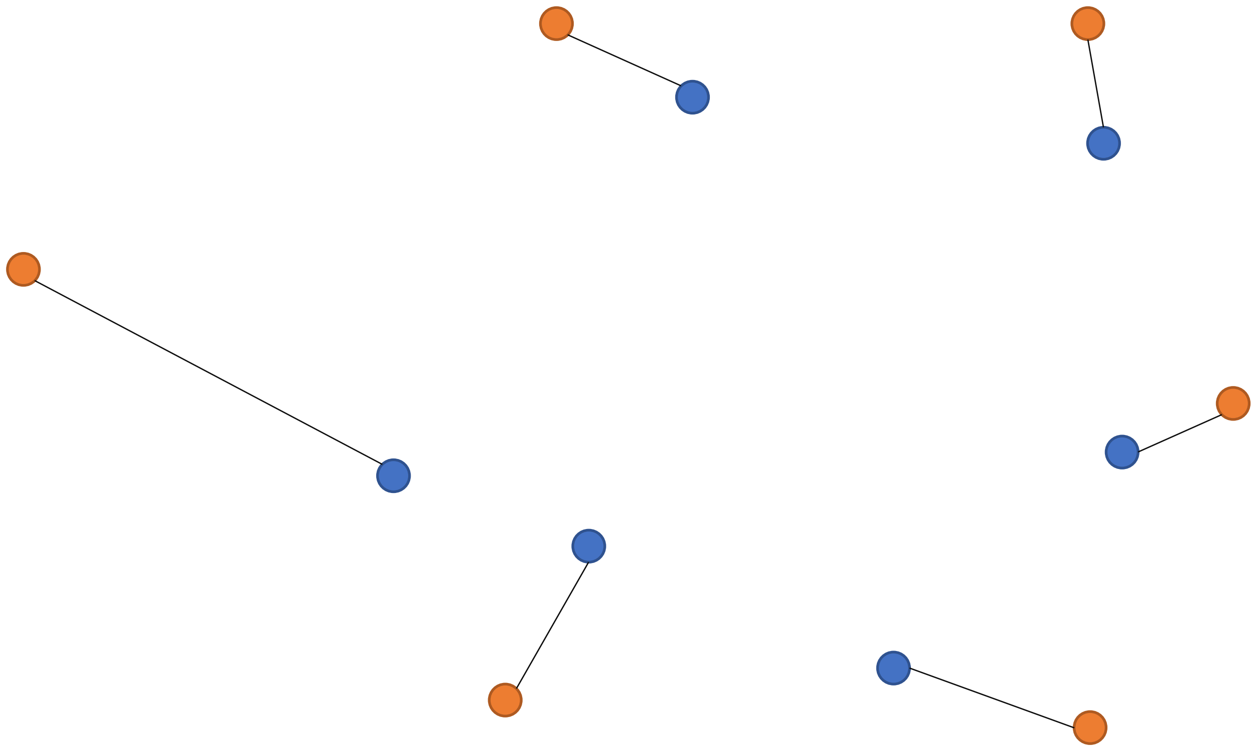
Worst case: Hausdorff distance (HD)

Average case: Chamfer distance (CD)

Optimal case: Earth Mover's distance (EMD)

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

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



Solves the optimal transportation (bipartite matching) problem!

Desired Properties of Distance Metrics

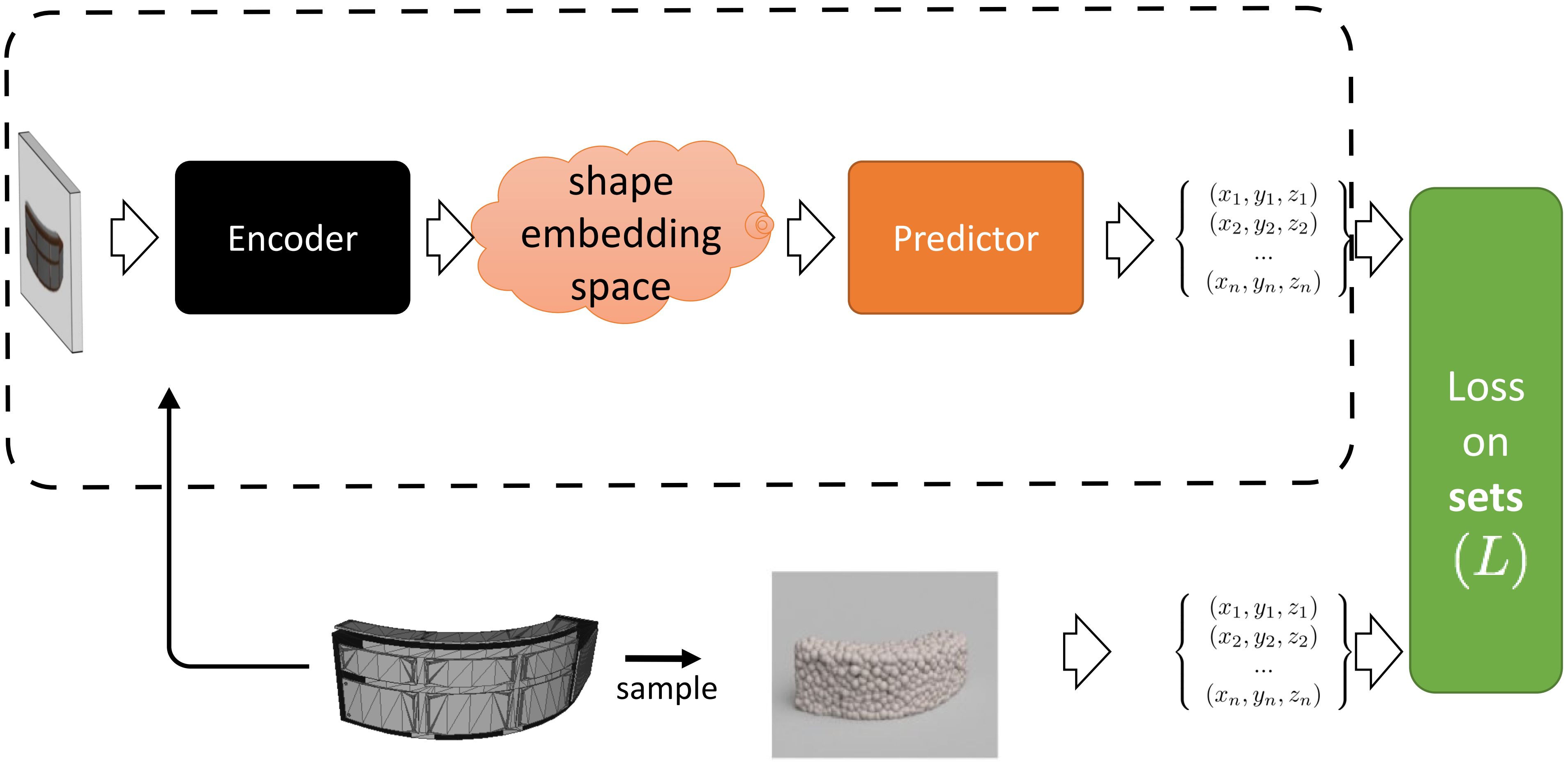
Geometric requirement

- Induces a nice shape space
- In other words, a good metric should reflect the natural shape differences

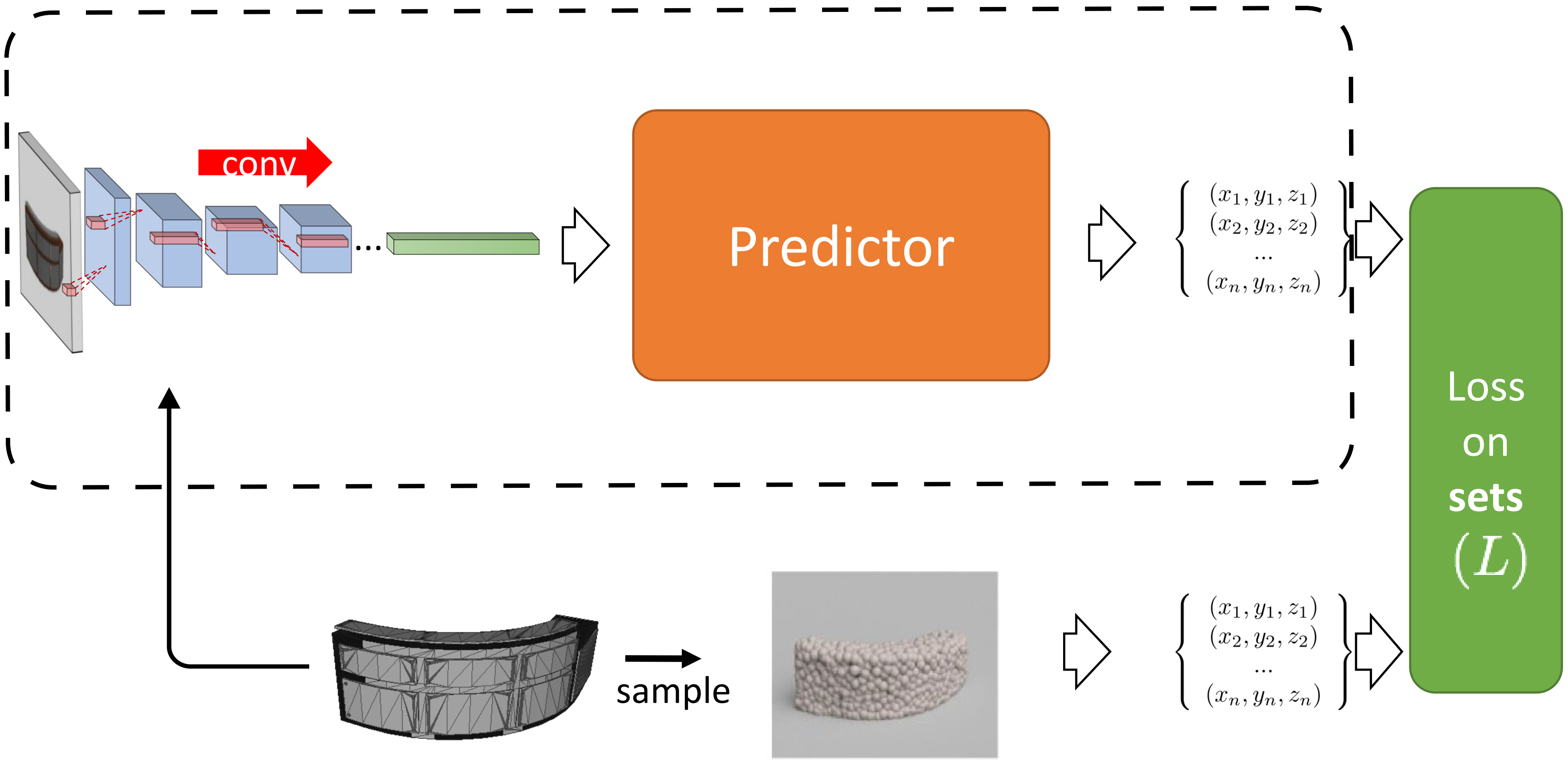
Computational requirement

- Defines a loss that is numerically easy to compute and optimize

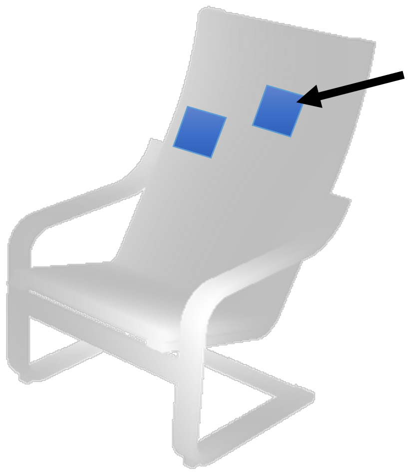
End-to-End Learning

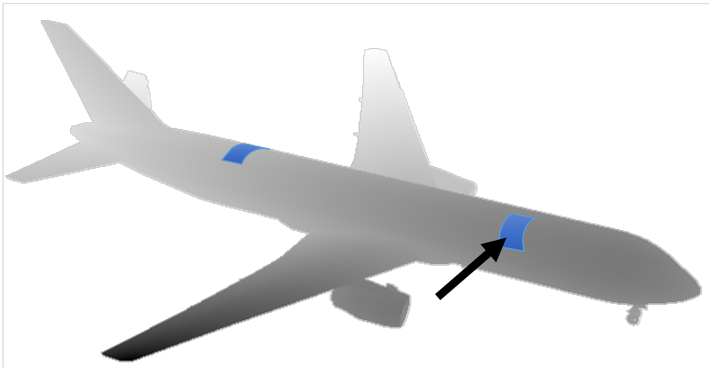


End-to-End Learning



Natural Statistics of Object Geometry

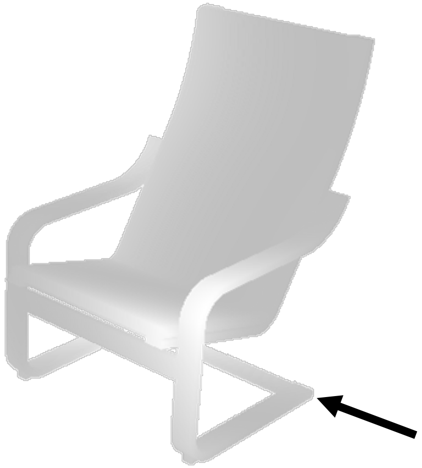


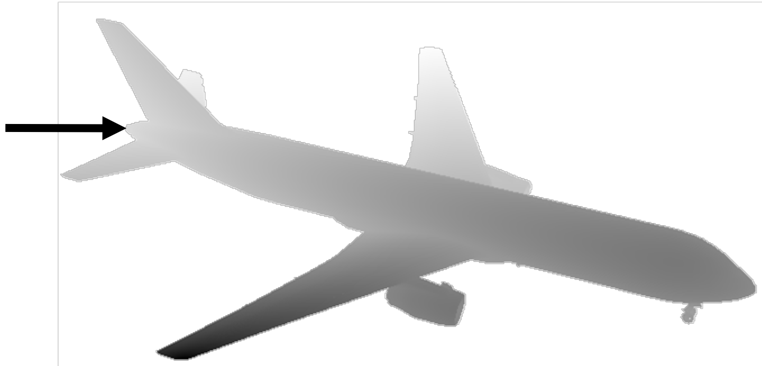


- Many local structures are common
  - e.g., planar patches, cylindrical patches
  - **strong local correlation** among point coordinates

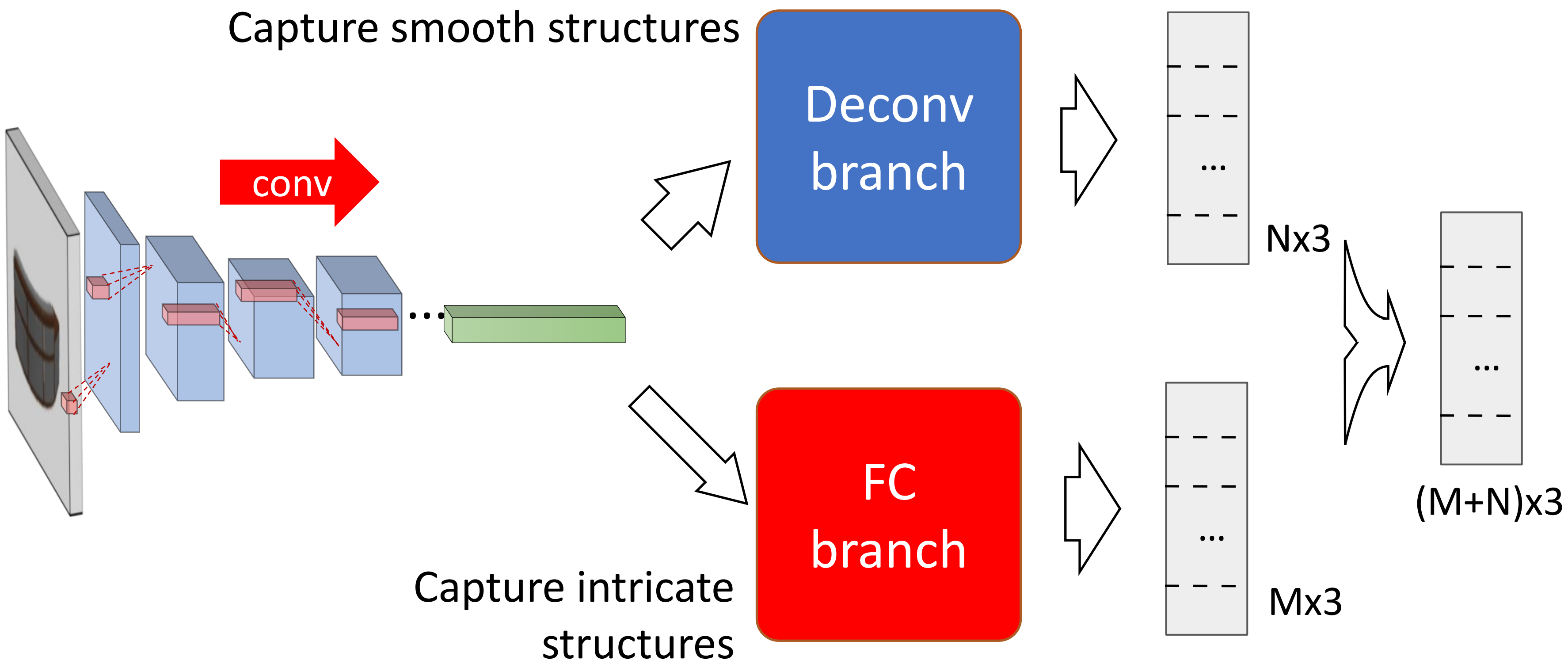
Natural Statistics of Object Geometry

- 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**





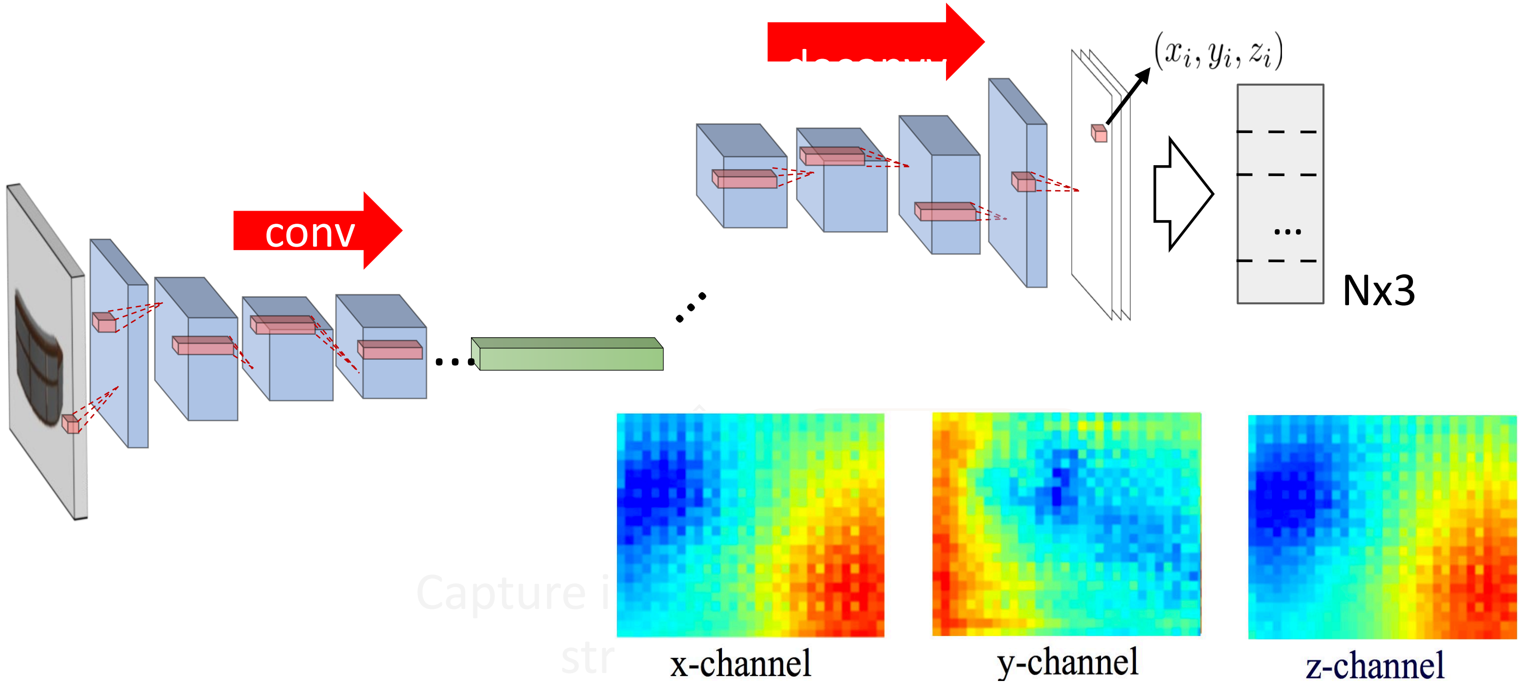
Two-Branch Architecture



Set union by array concatenation

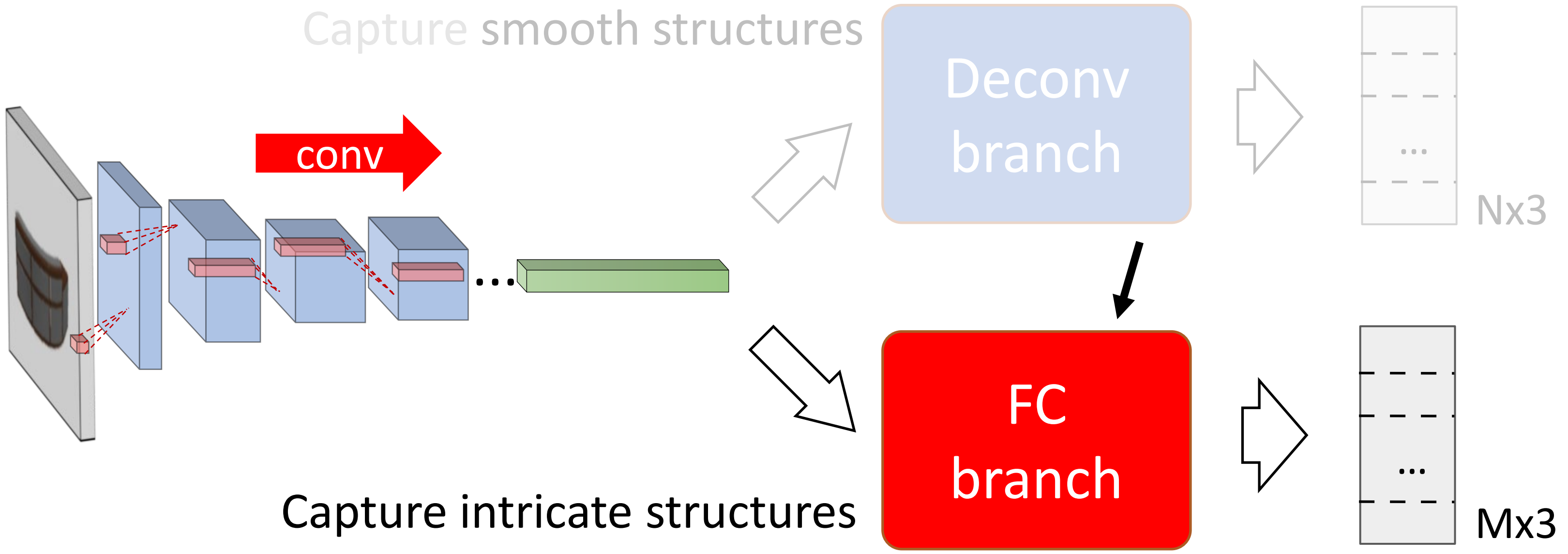
Deconvolution Branch

Parametrization / coordinate map



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

Fully Connected Branch



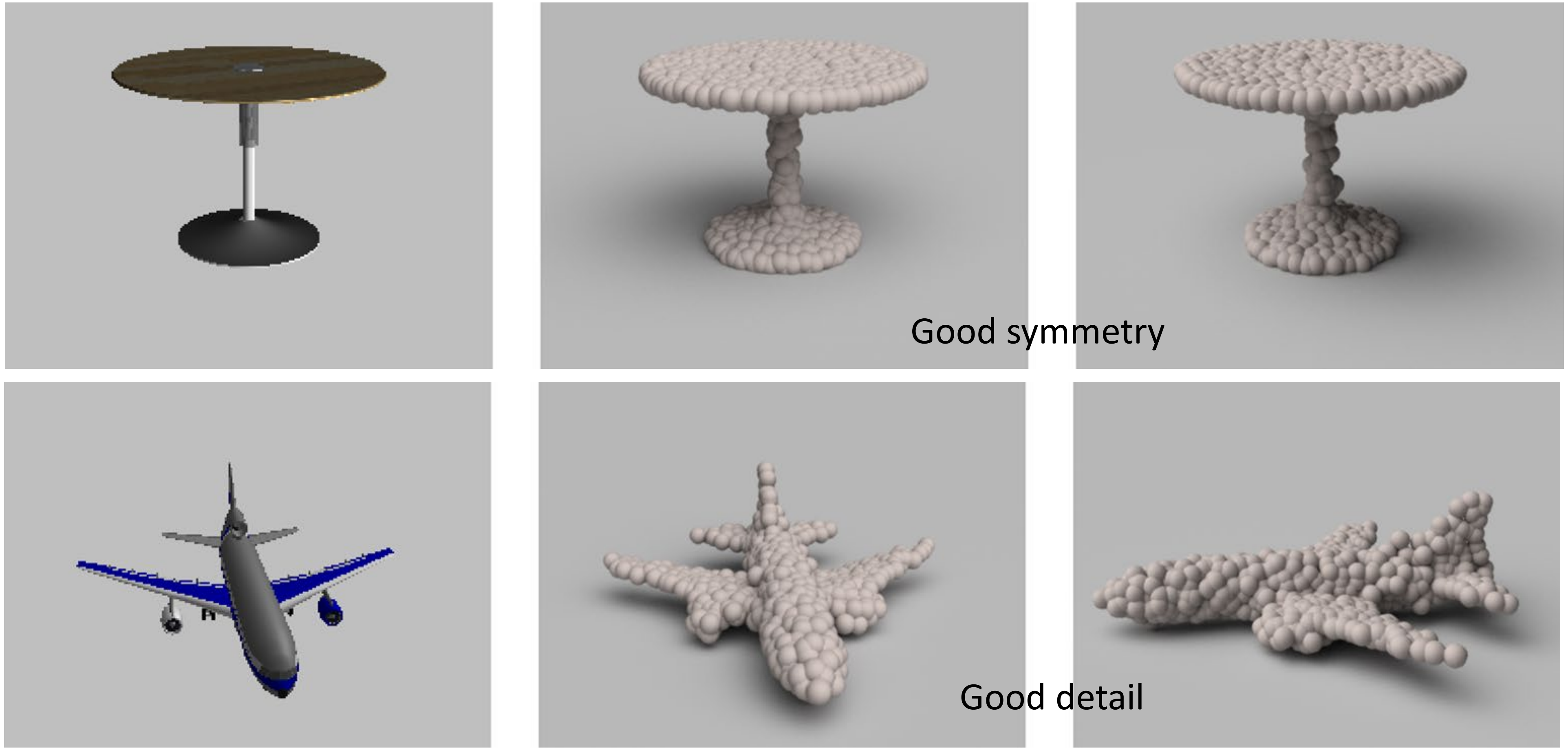
The Two Branches

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

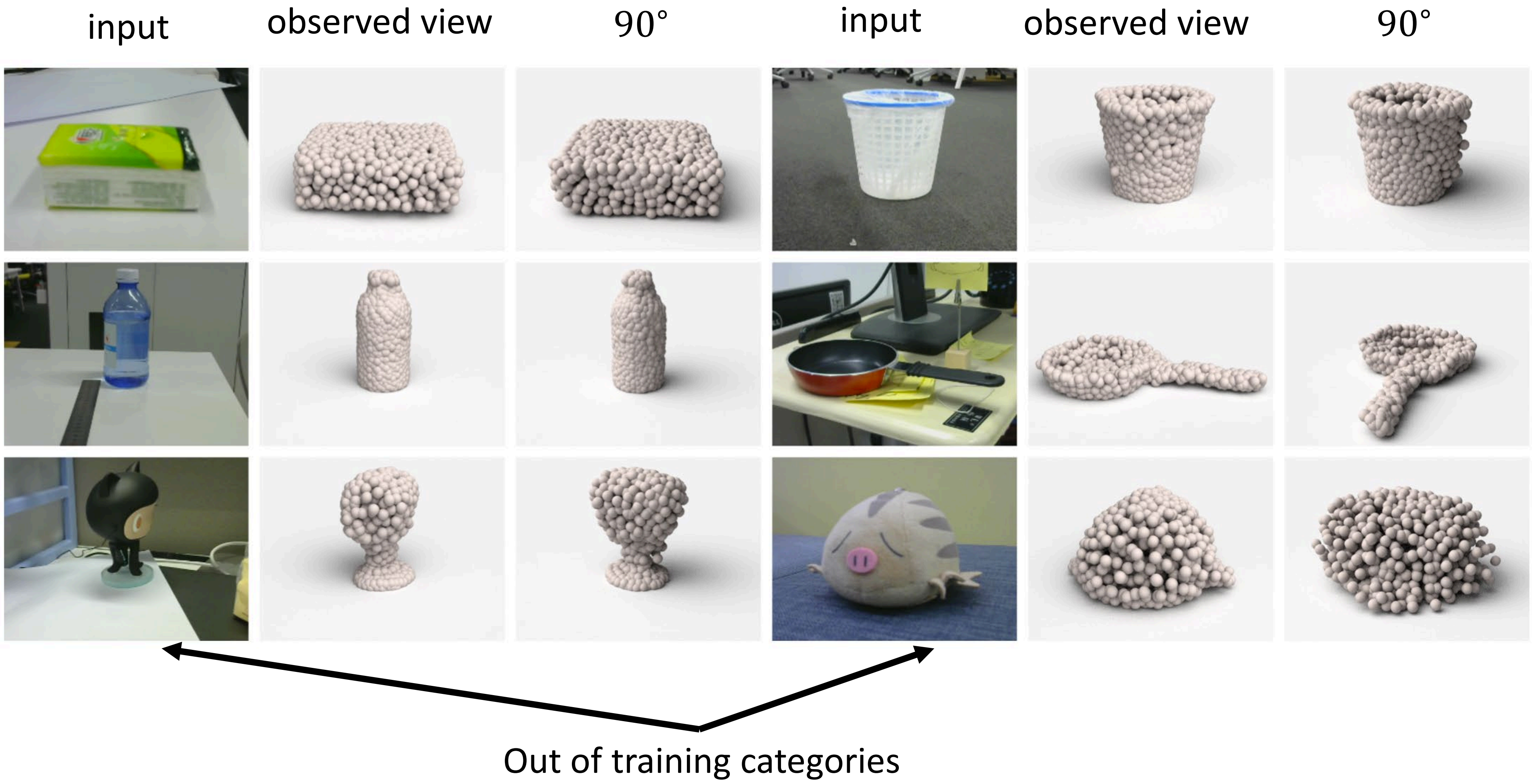
red: fully-connected branch – **more intricate** structures

Example Results

Same view New view



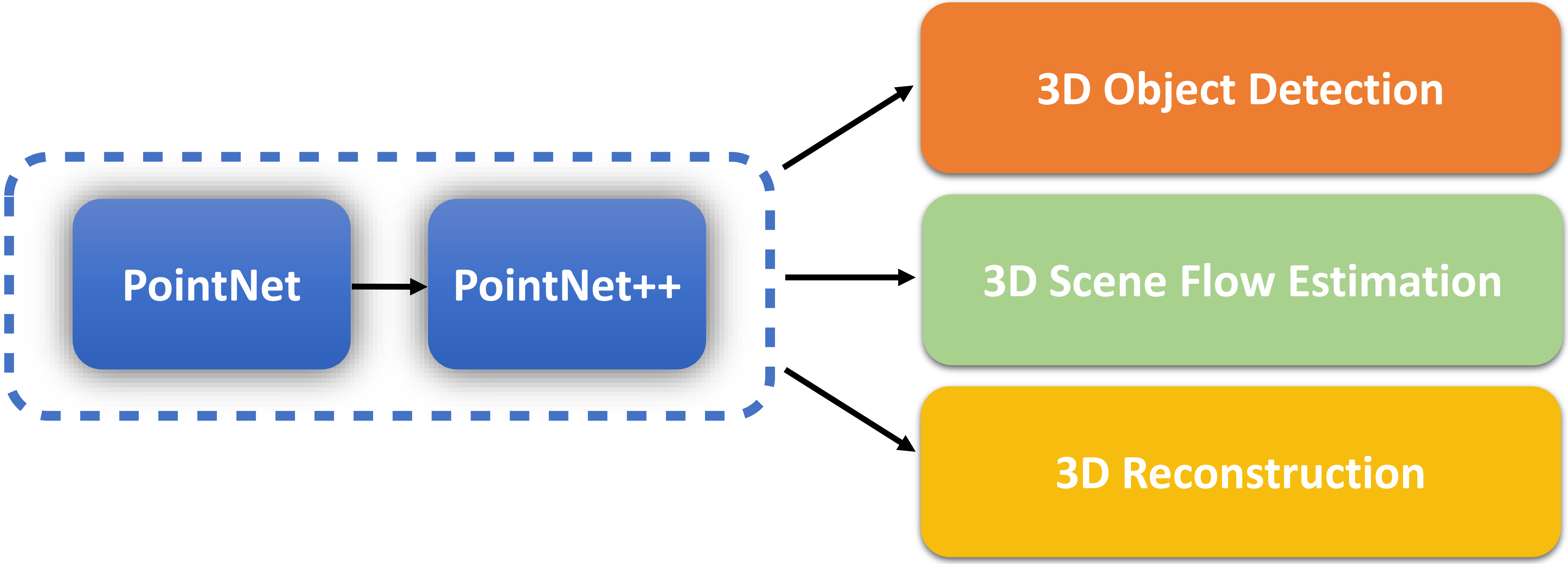
From Real Images



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