Recognition in Point Clouds

3D Vision

University of Illinois

Derek Hoiem

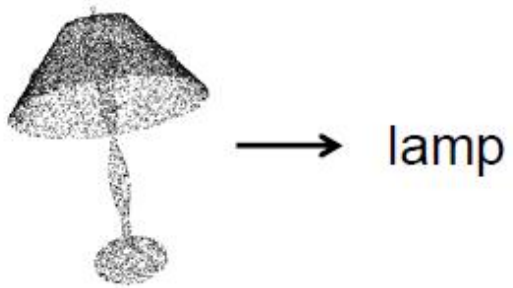
Survey

- Due tonight: put link to pdf on Google drive
- Sign up to review another survey
  - Up to 4 reviewers per survey
  - See assignment for what to address, 100+ words
  - Add link to Google doc
- Question: on Tues have survey groups present or 3D recognition papers?

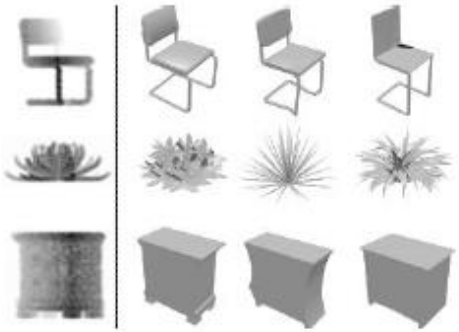
This class: Recognition in Point Clouds

- Problem domain overview
- PointNet / PointNet++
- Octree-based O-CNN
- MinkowskiNet
- 2D/3D BPNet

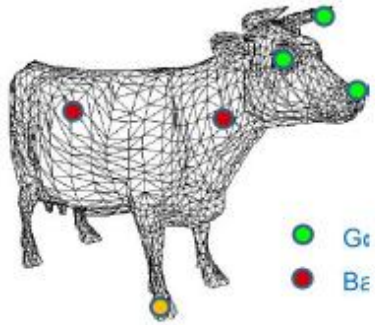
Tasks



shape classification

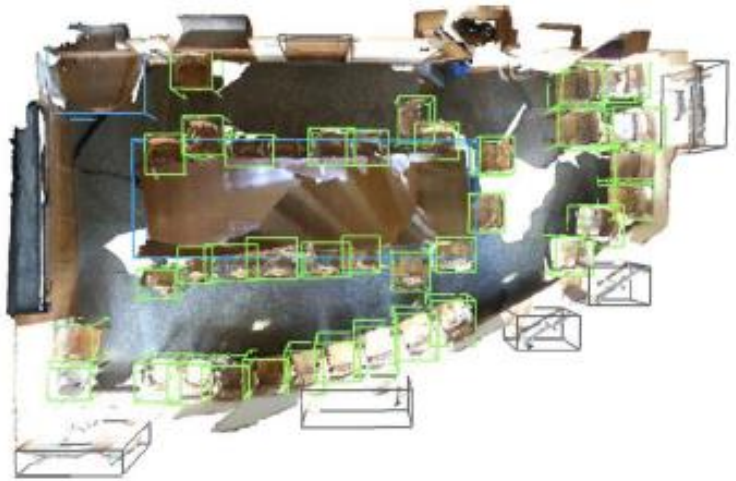


shape retrieval



keypoint detection

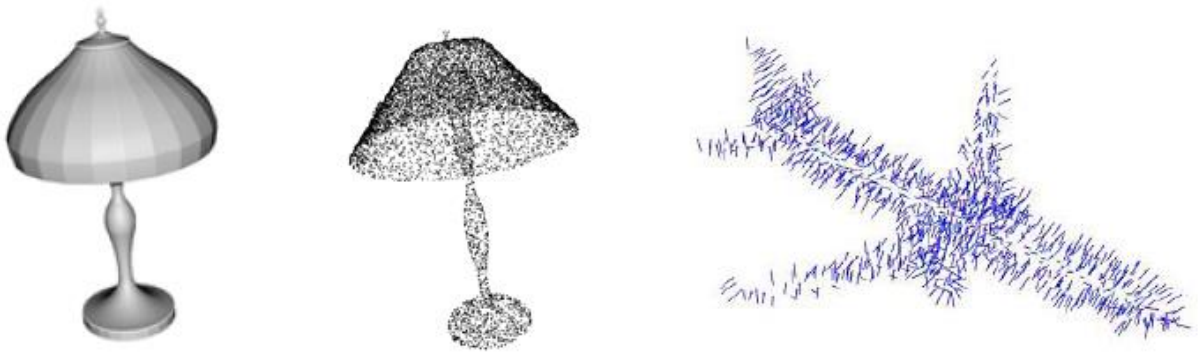
semantic segmentation



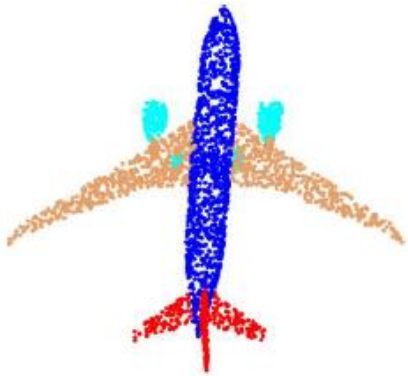
object detection

Slide Credit: Yongcheng Liu

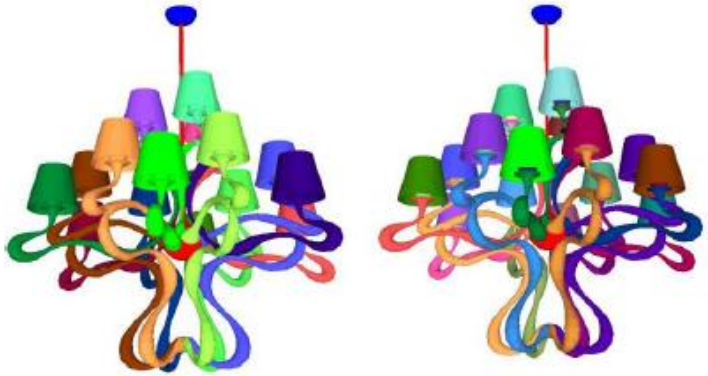
Datasets



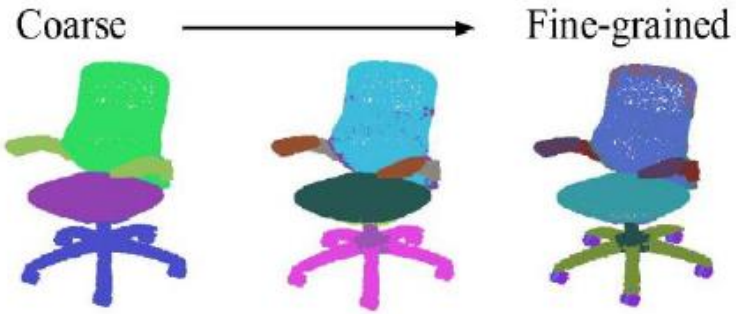
Princeton ModelNet: 1k



ShapeNet Part: 2k



PartNet models



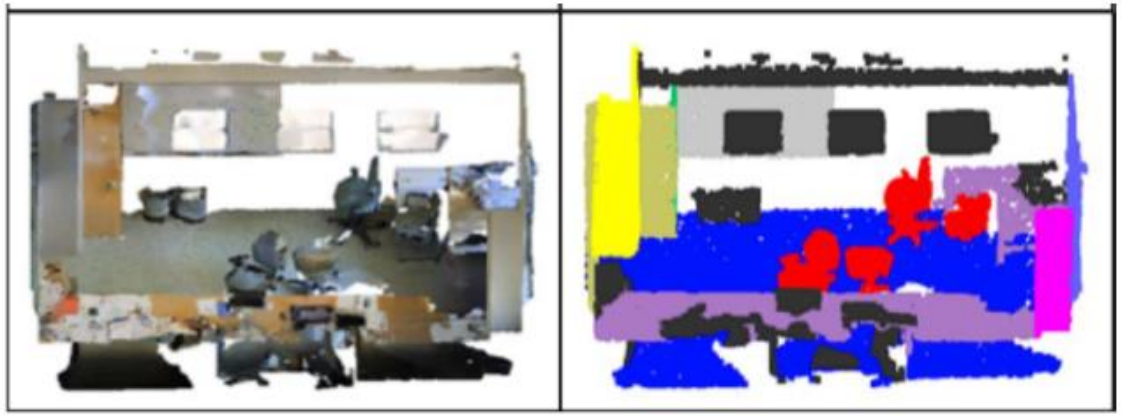
Hierarchical Semantic Segmentation

.....

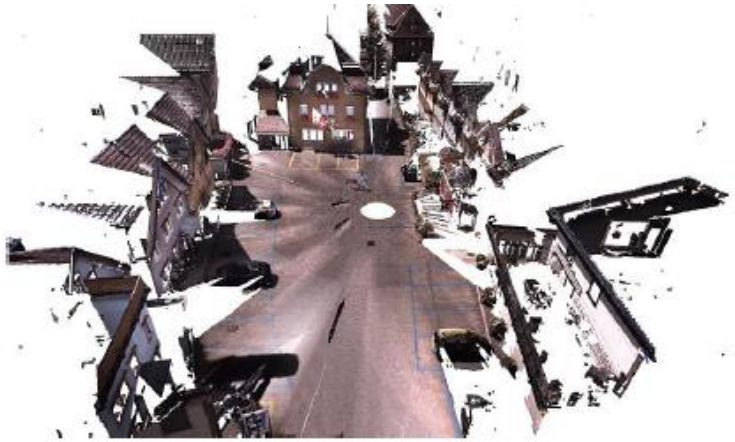
Mo et al. PartNet: A Large-scale Benchmark for Fine-grained and Hierarchical Part-level 3D Object Understanding. CVPR 2019.
Yi et al. A scalable active framework for region annotation in 3D shape collections. TOG 2016.
Wu et al. 3D ShapeNets: A Deep Representation for Volumetric Shapes. CVPR 2015.

Slide Credit: Yongcheng Liu

Datasets

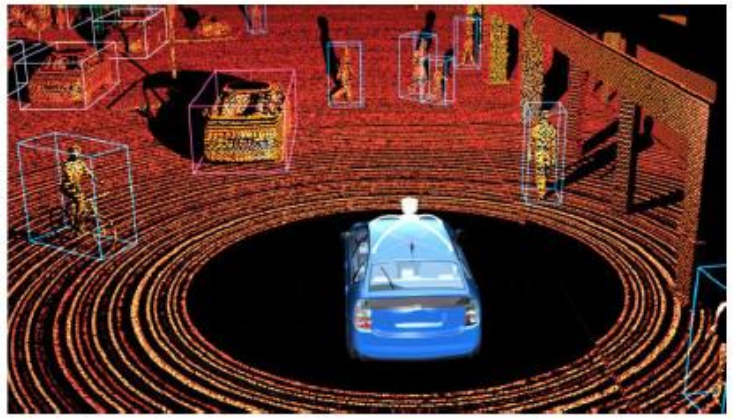


Stanford 3D indoor scene: 8k



Semantic 3D: 4 billion in total

ScanNet: seg + det



KITTI: det

.....

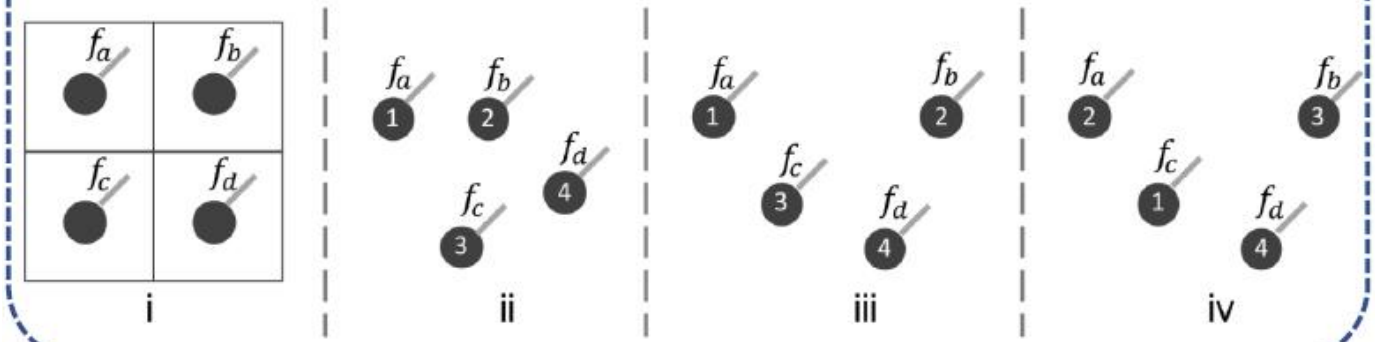
Dai et al. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. CVPR 2017.
Armeni et al. 3d semantic parsing of large-scale indoor spaces. CVPR 2016.
Hackel et al. Semantic3d. net: A new large-scale point cloud classification benchmark. ISPRS 2017.

Slide Credit: Yongcheng Liu

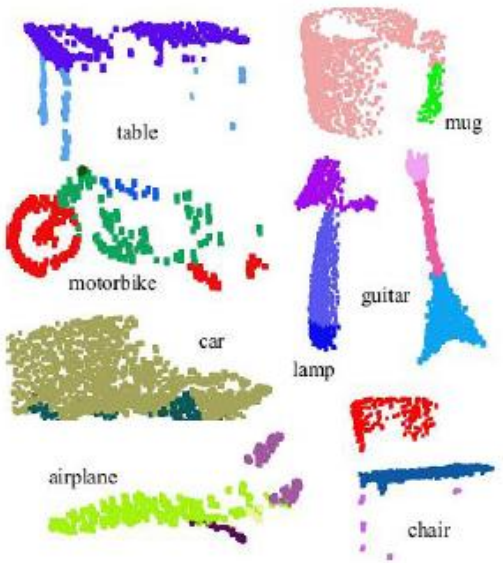
Slide Credit: Yongcheng Liu

General challenges

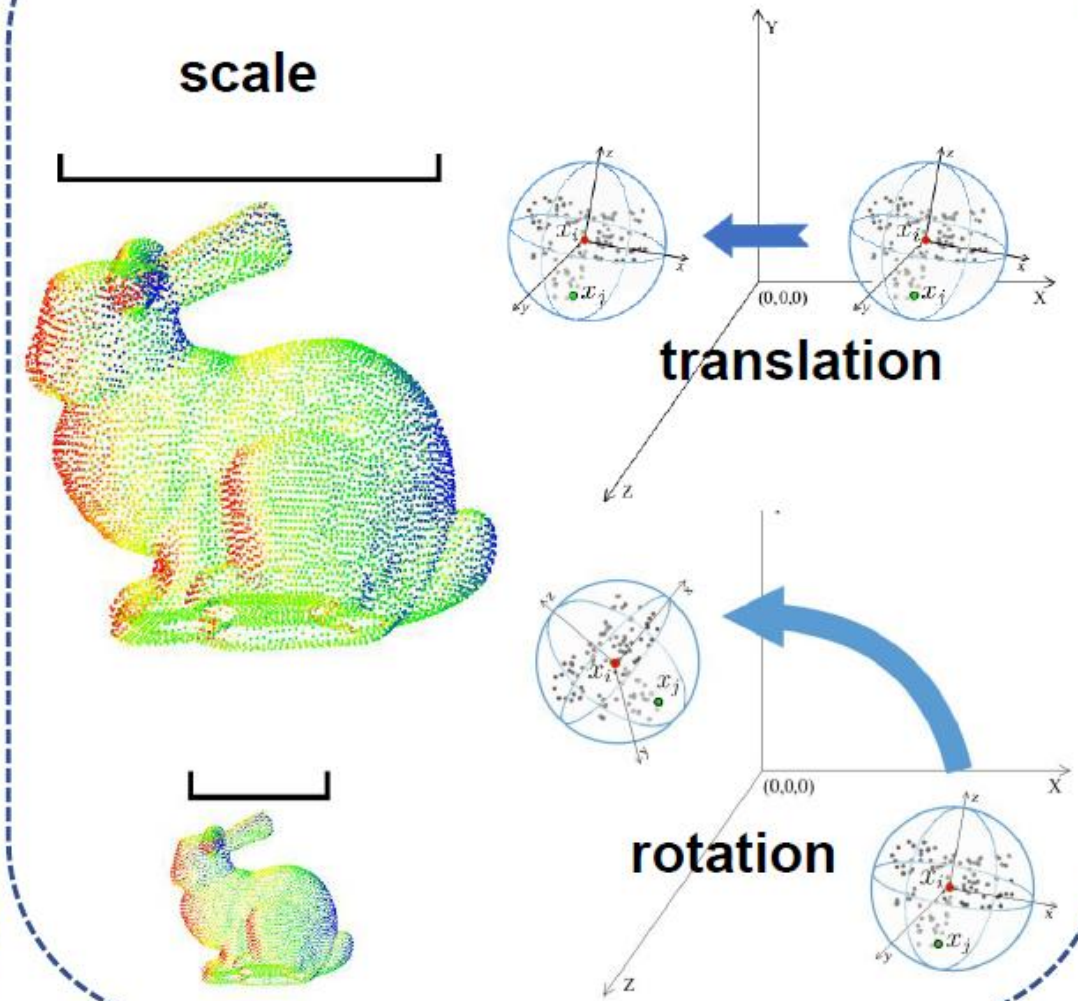
Irregular (unordered): permutation invariance



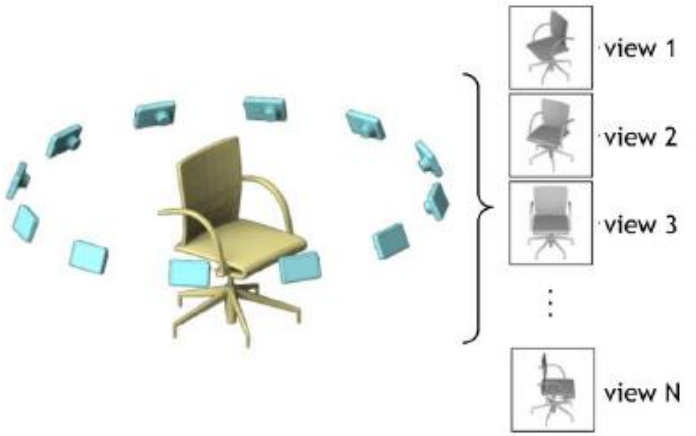
Robustness to corruption, outlier, noise; partial data



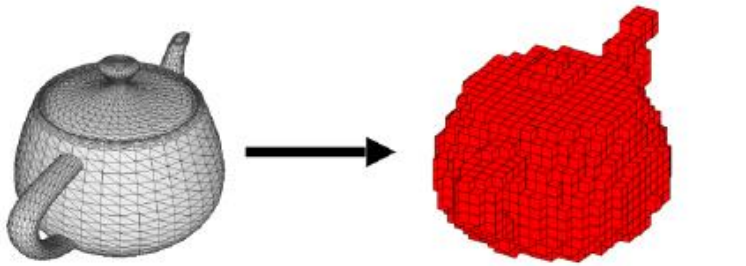
Robustness to rigid transformations



Representations



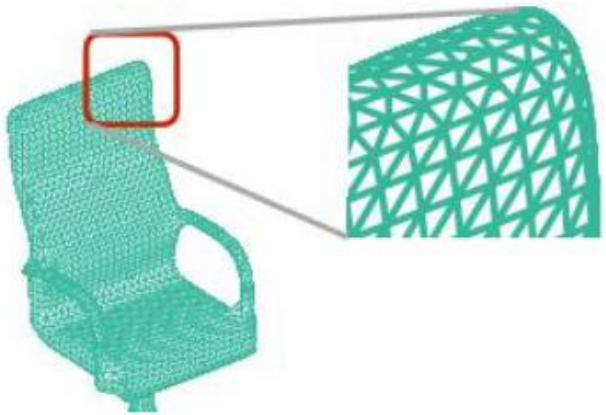
multi-view images + 2D CNN



CAD model

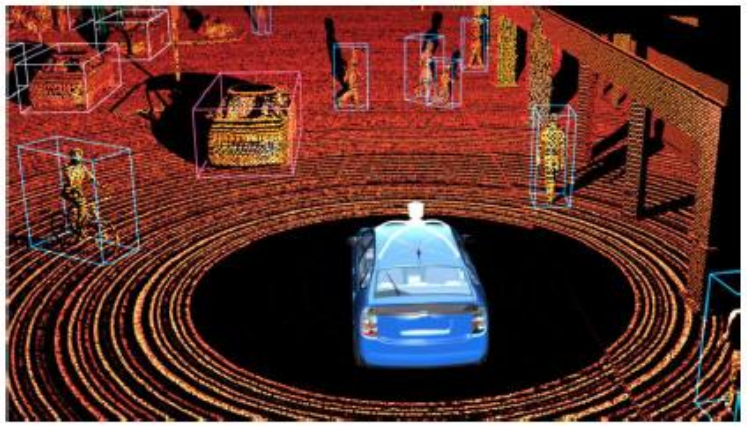
Occupancy Grid
30x30x30

volumetric data + 3D CNN



mesh data + DL (GNN) ?

image depth + CNN

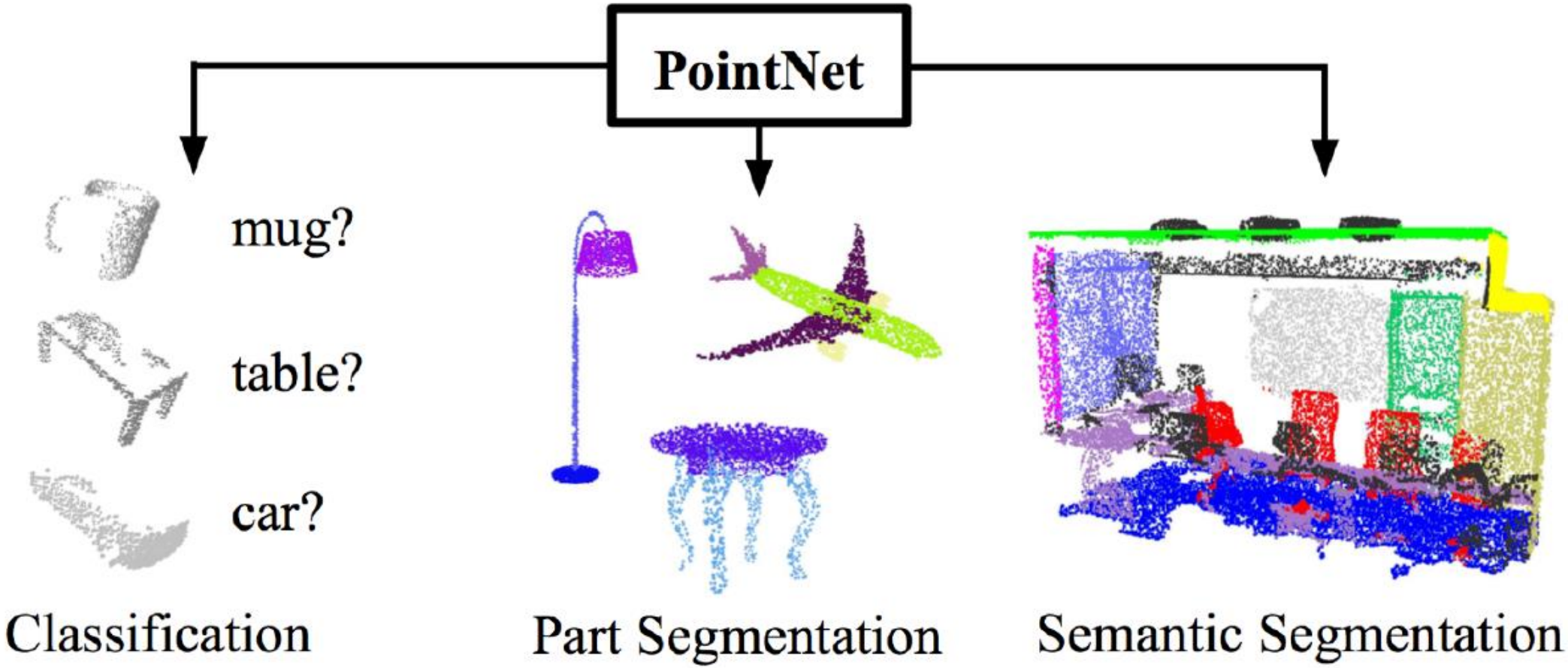


point cloud + DL (CNN) ?

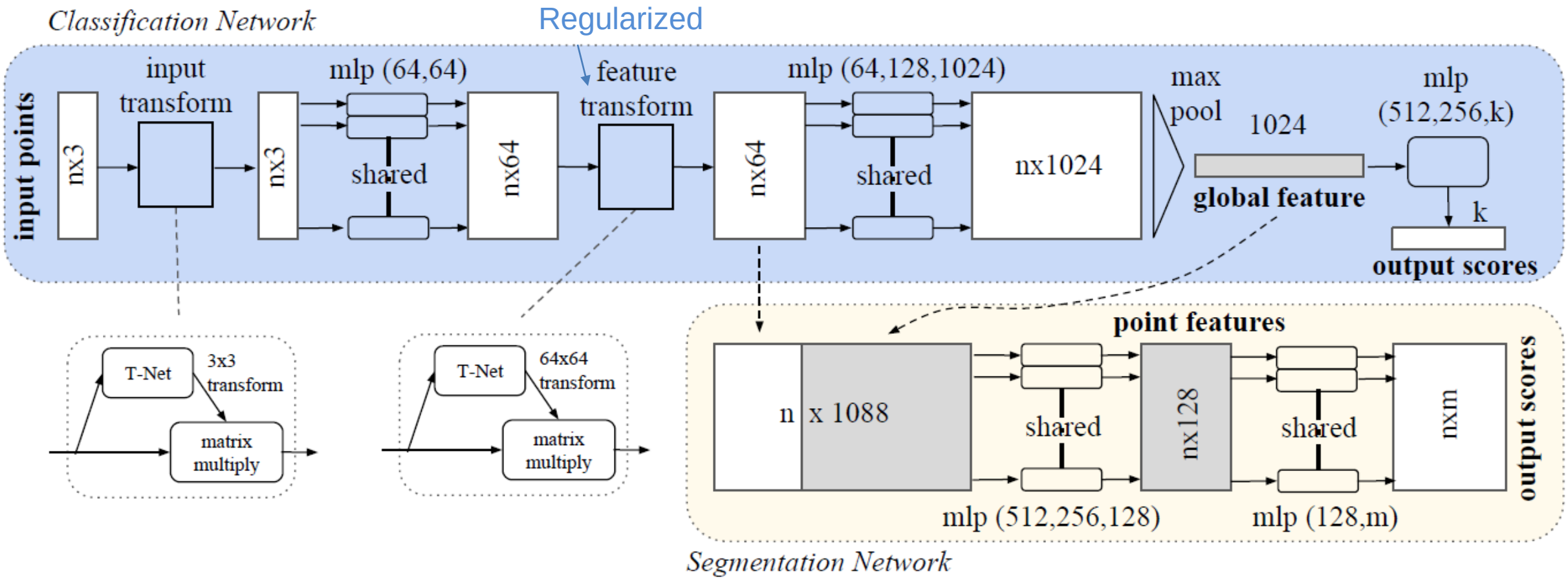
Slide Credit: Yongcheng Liu

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

Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas
Stanford University



CVPR 2017



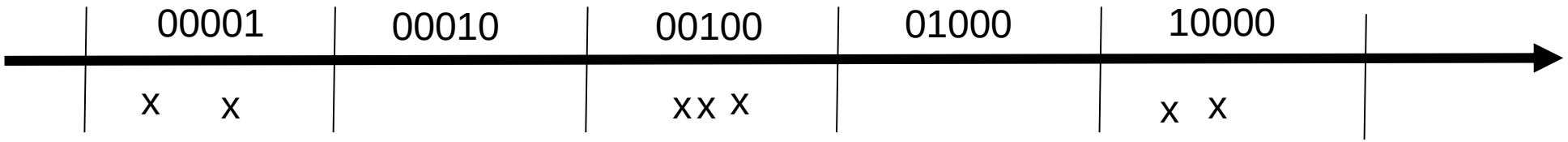
- Unordered point set as input
  - Transform (point → feature) and then max pool. For example, each point could be mapped to a feature that encodes a voxel and then the global feature would represent which voxels are filled.
  - Point transformations are independent of other points!

- Robust to geometric transformations
  - Predicted 3x3 transformation enables point cloud to be transformed before processing

Simple 1D invariance example

Seven 1-dim points (x's)

Bins = mapping into 5-dim features



Max pooling

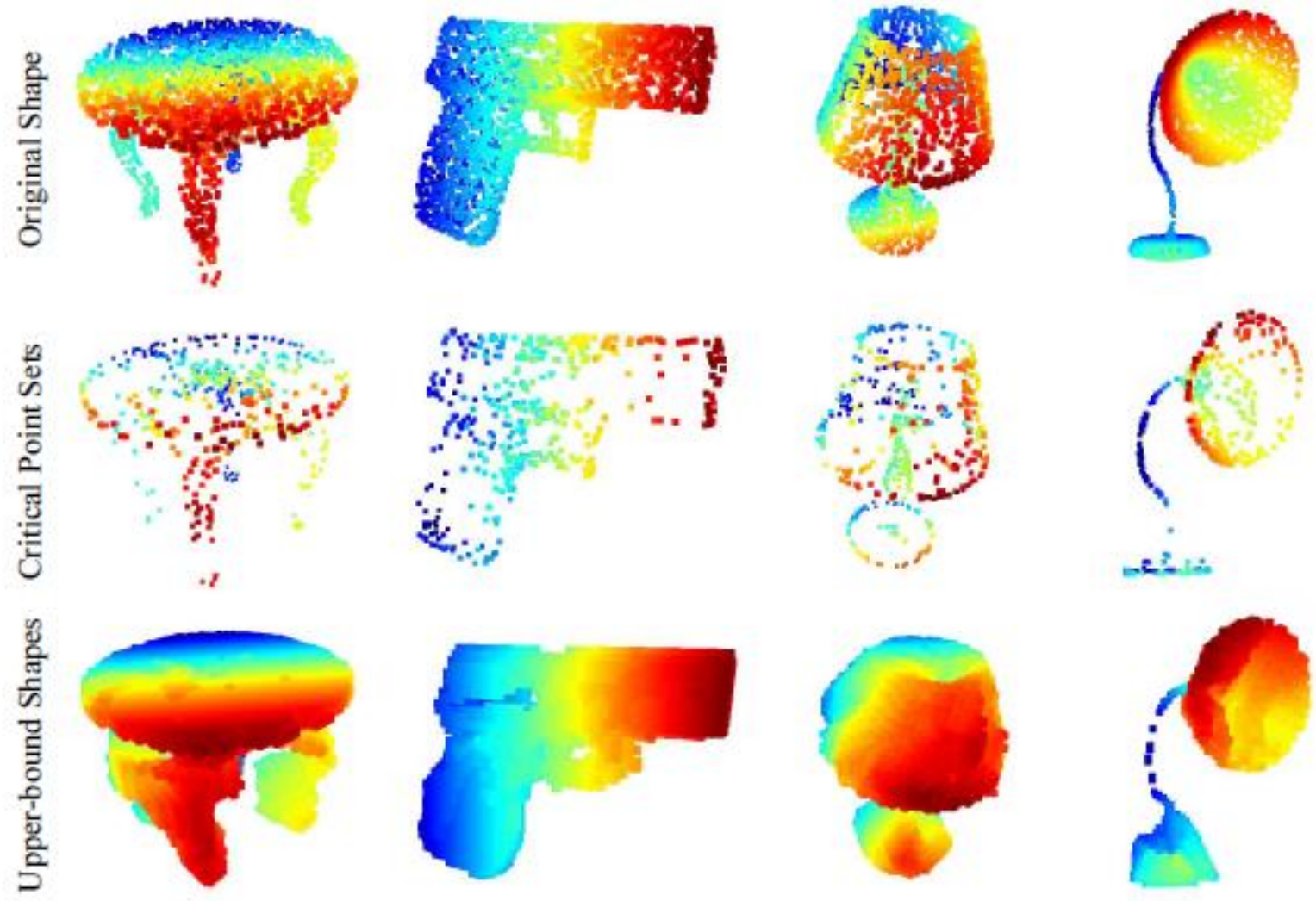
10101

In practice, mapping from 3D points to 1024D features does not need to be as simple as a partitioning into 1024 cells, and the mapping is learned

Robust to varying point density

Minimum set of points to get same features

Maximum set of points to get same features



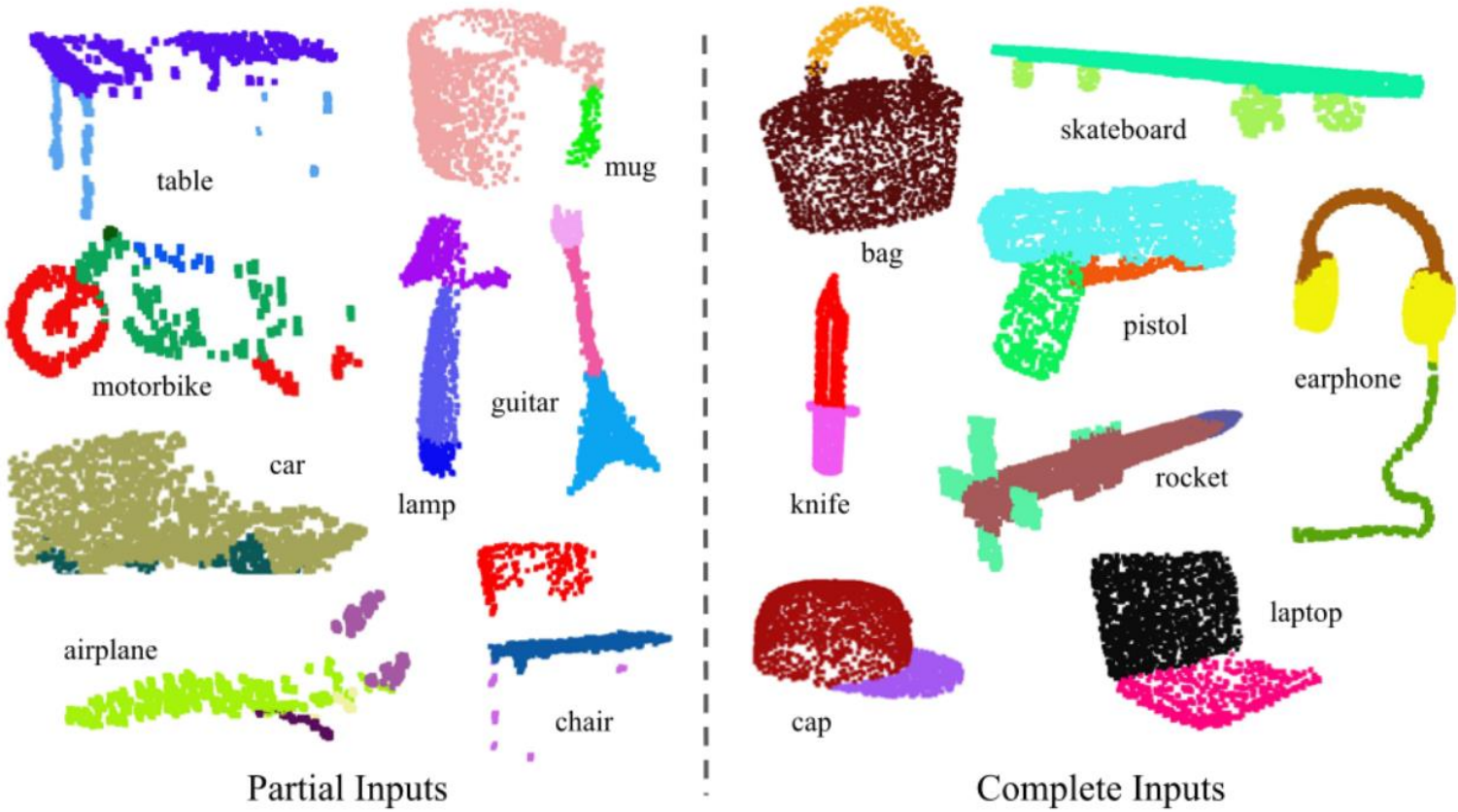
Results on Object Classification

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

3D CNNs

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

Results on Object Part Segmentation



Results on Object Part Segmentation

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

dataset: ShapeNetPart; metric: mean IoU (%)

Results on Semantic Scene Parsing

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

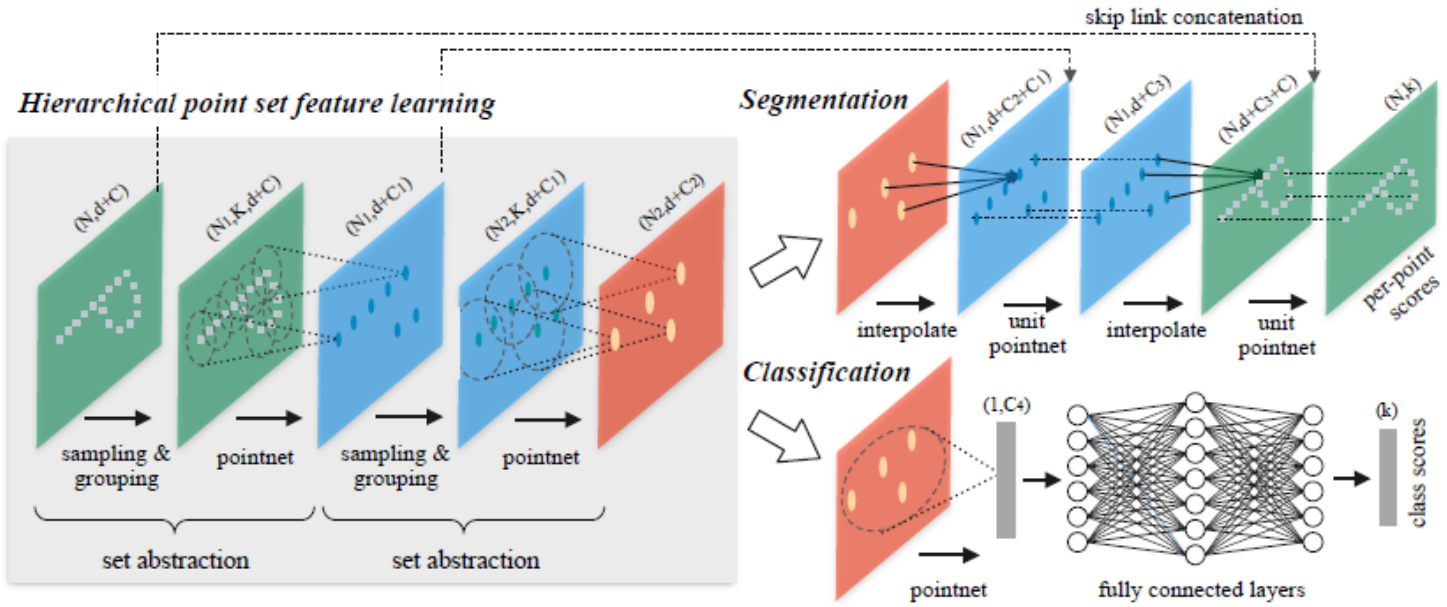
PointNet pros and cons

+ Process 1 million points per second on GTX1080 GPU
+ Can incorporate many features (position, color, normal, local shape)
+ Many applications: object classification, point labeling, point normal estimation, retrieval, keypoint matching

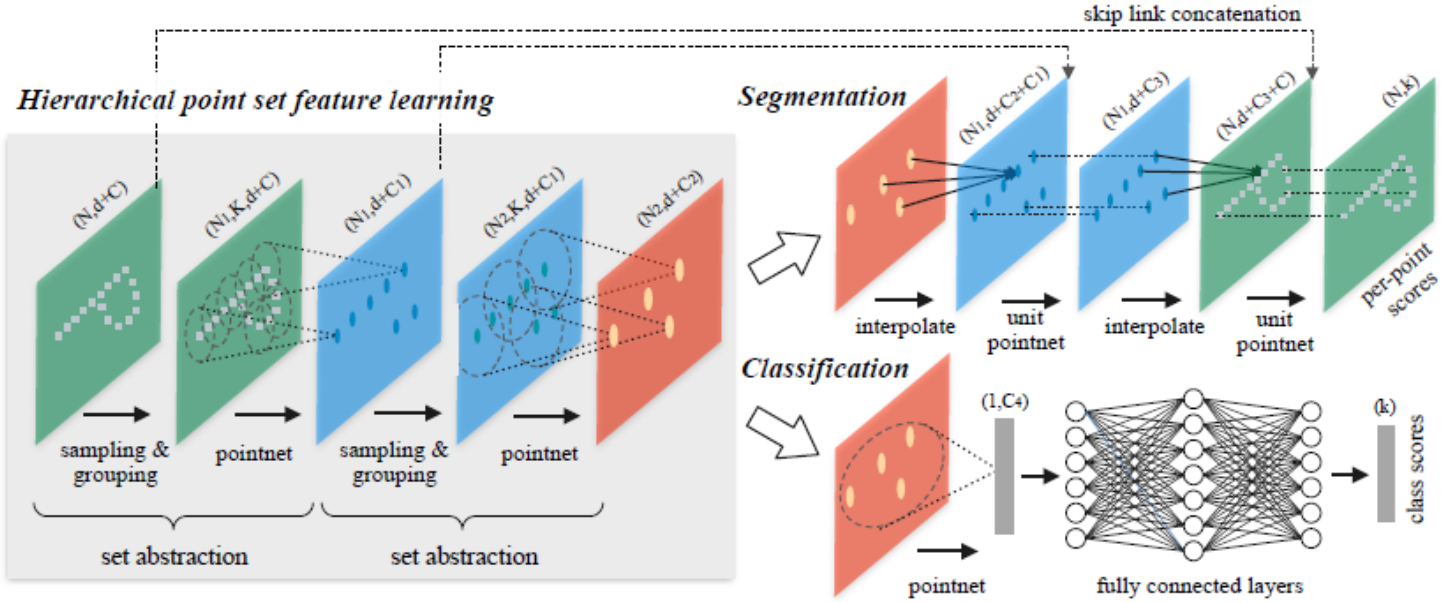
- Limited resolution (due to 1024 global vector)
- Cannot learn local shape features other than occupancy
- Not as accurate as subsequent methods

PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space

Charles R. Qi Li Yi Hao Su Leonidas J. Guibas
Stanford University



NeurIPS 2017



- **Sampling**: iterative farthest point to get N' cluster centers
- **Grouping**: Points within ball radius of each cluster center are selected
- **PointNet**: maps each point from d+C to d+C' dimensions and maxpools (encode occupancy of local neighborhood)

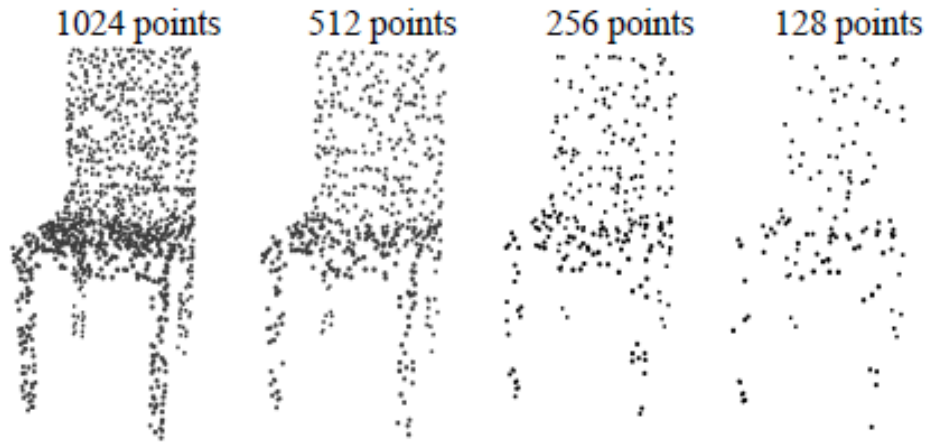
- Multi-scale Grouping: apply grouping with different radii in parallel and concatenate; train with point dropout
- Feature propagation: in skip links, interpolate feature values and concatenate with each point; then pass through 1x1 conv

Method	Error rate (%)
Multi-layer perceptron [24]	1.60
LeNet5 [11]	0.80
Network in Network [13]	**0.47**
PointNet (vanilla) [20]	1.30
PointNet [20]	0.78
Ours	0.51

Table 1: MNIST digit classification.

Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9**

Table 2: ModelNet40 shape classification.



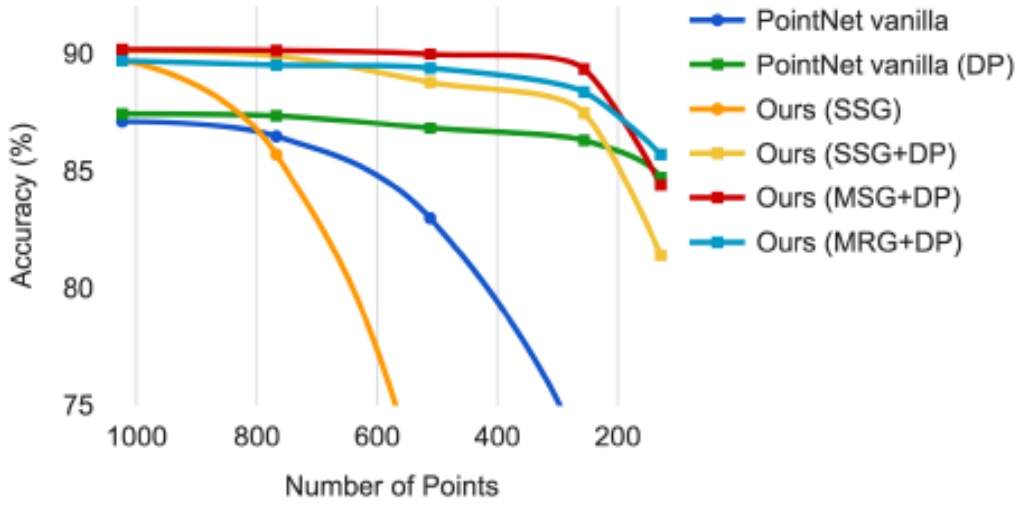


Figure 4: Left: Point cloud with random point dropout. Right: Curve showing advantage of our density adaptive strategy in dealing with non-uniform density. DP means random input dropout during training; otherwise training is on uniformly dense points. See Sec.3.3 for details.

- Uses 4 layers of PointNet (vs. 3 for classification)
- Operates on a 3x2.5x2.5 m volume of points at a time

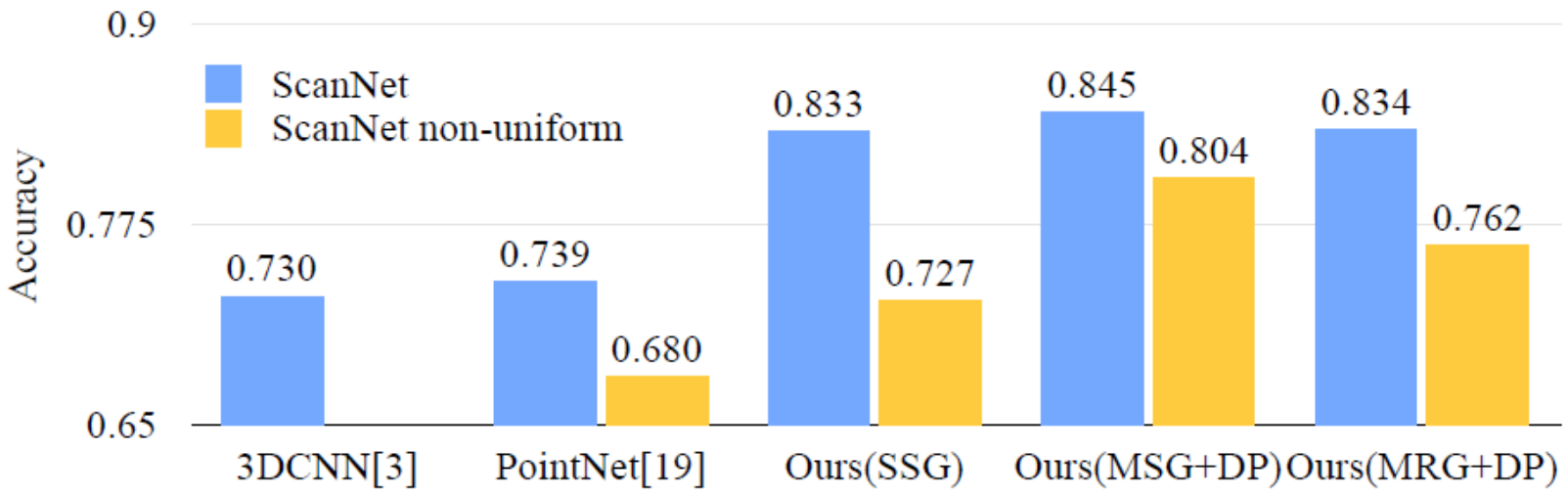


Figure 5: Scannet labeling accuracy.

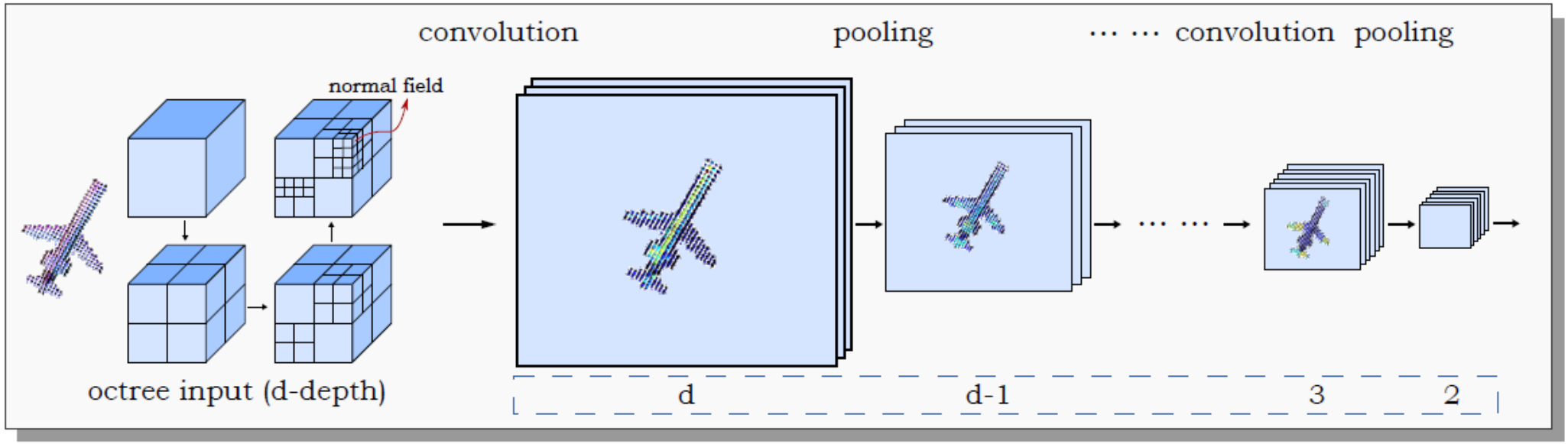
PointNet++ pros and cons

+ Improves accuracy: clustering approach enables computing features of local geometry

- More complicated than PointNet: more hyperparamters parameters and more variations between application settings
- 3x (or more) slower than PointNet
- Does not address resolution problem

O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis

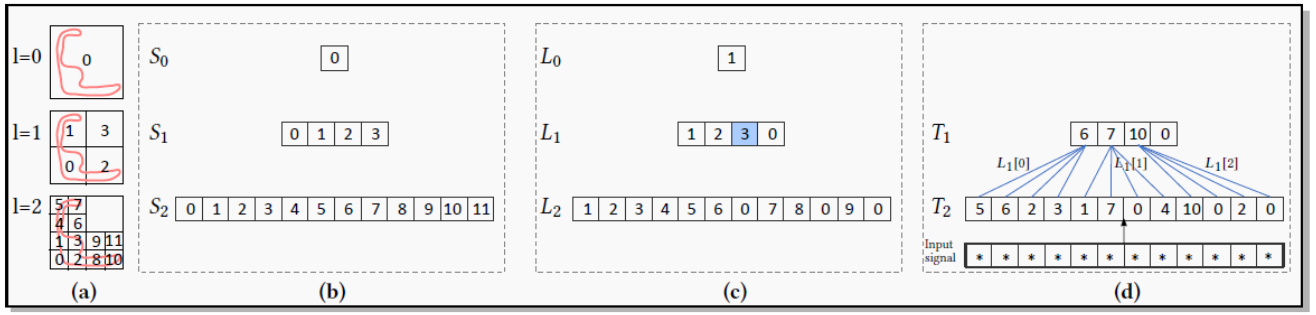
PENG-SHUAI WANG, Tsinghua University and Microsoft Research Asia
YANG LIU, Microsoft Research Asia
YU-XIAO GUO, University of Electronic Science and Technology of China and Microsoft Research Asia
CHUN-YU SUN, Tsinghua University and Microsoft Research Asia
XIN TONG, Microsoft Research Asia



SIGGRAPH 2017

O-CNN

- Exploits Octree sparsity and organization to create efficient data structures for 3D convolution (this is the clever part)
- Store average point normals in leaf nodes of Octree – only compute over occupied nodes
- $O(n^2)$ space/time for n resolution, compared to $O(n^3)$ for voxels
- Simple architecture: multiple convolution + batch-norm + relu + pool modules, followed by FC layers
- For point labeling, encoding is followed by upsampling decoder, similar to UNet



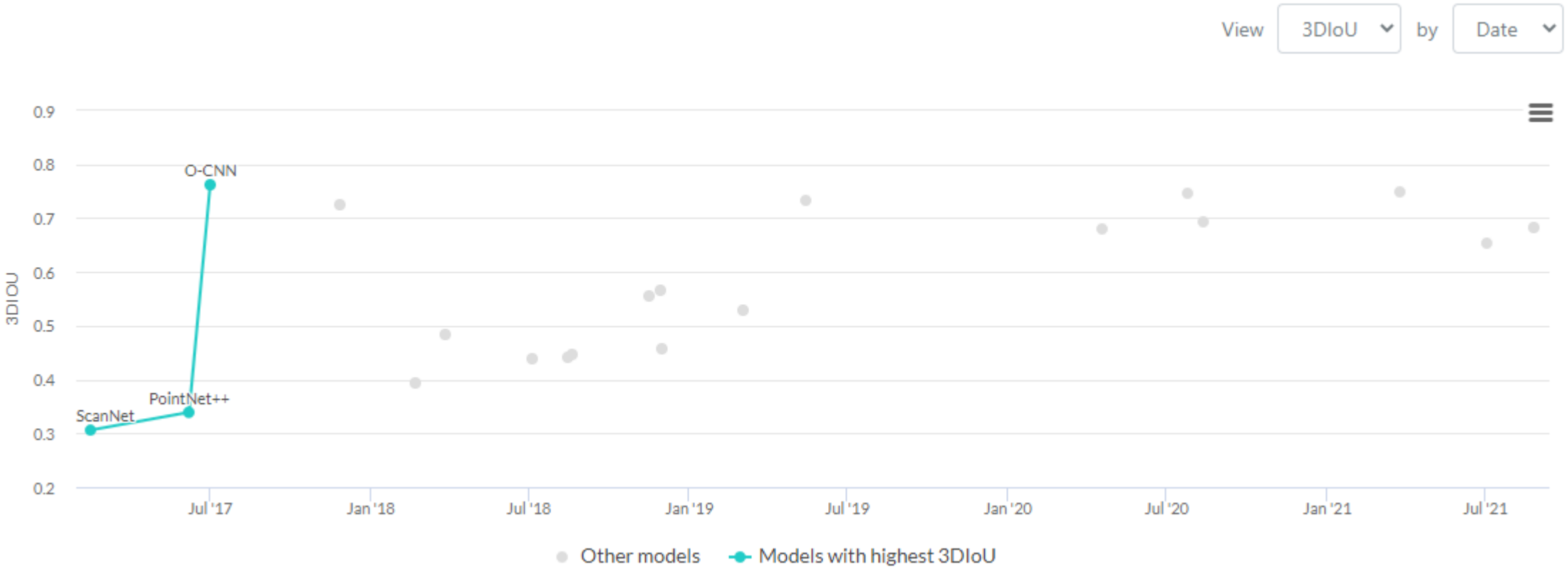
Method	16^3	32^3	64^3	128^3	256^3
O-CNN	0.32GB	0.58GB	1.1GB	2.7GB	6.4GB
full voxel+binary	0.23GB	0.71GB	3.7GB	Out of memory	Out of memory
full voxel+normal	0.27GB	1.20GB	4.3GB	Out of memory	Out of memory

Table 3. Comparisons on GPU-memory consumption. The batch size is 32.

Method	16^3	32^3	64^3	128^3	256^3
O-CNN	17ms	33ms	90ms	327ms	1265ms
full voxel+binary	59ms	425ms	1648ms	-	-
full voxel+normal	75ms	510ms	4654ms	-	-

Table 4. Timings of one backward and forward operation in milliseconds. The batch size is 32.

O-CNN – best performing in ScanNet(!?)



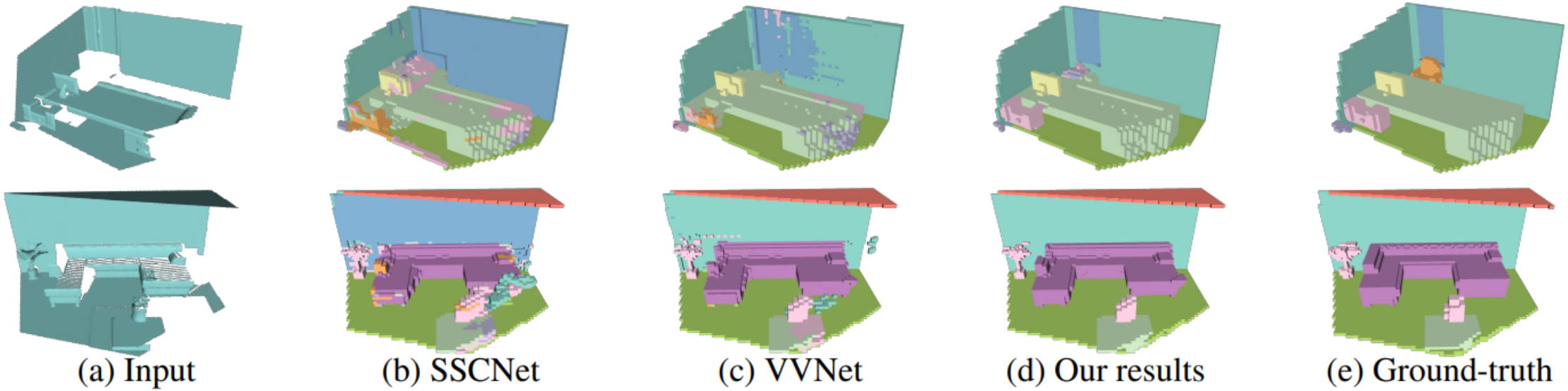
What's New?

- 2021.08.24: Update the code for pythorch-based O-CNN, including a UNet and some other major components. Our vanilla implementation without any tricks on ScanNet dataset achieves 76.2 mIoU on the ScanNet benchmark, even surpassing the recent state-of-art approaches published in CVPR 2021 and ICCV 2021.

https://github.com/Microsoft/O-CNN

Extension to scene completion (CVPR 2020 workshop)

- Complete occluded portion of depth image



	Scene completion			Semantic scene completion											
Method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
3DRecGAN [46]	-	-	72.1	79.9	75.2	48.2	28.9	20.2	64.4	54.6	25.7	17.4	33.7	24.4	43.0
SSCNet [32]	76.3	95.2	73.5	96.3	84.9	56.8	28.2	21.3	56.0	52.7	33.7	10.9	44.3	25.4	46.4
ForkNet [40]	-	-	86.9	95.0	85.9	73.2	54.5	46.0	81.3	74.2	42.8	31.9	63.1	49.3	63.4
SATNet [20]	80.7	96.5	78.5	97.9	82.5	57.7	58.5	45.1	78.4	72.3	47.3	45.7	67.1	55.2	64.3
VVNet [10]	90.8	91.7	84.0	98.4	87.0	61.0	54.8	49.3	83.0	75.5	55.1	43.5	68.8	57.7	66.7
SGCNet [48]	92.6	90.4	84.5	96.6	83.7	74.9	59.0	55.1	83.3	78.0	61.5	47.4	73.5	62.9	70.5
CCPNet [49]	98.2	96.8	91.4	**99.2**	89.3	76.2	**63.3**	58.2	86.1	**82.6**	65.6	53.2	**76.8**	65.2	74.2
Our Results	92.1	95.5	88.1	98.2	**92.8**	**76.3**	61.9	**62.4**	**87.5**	80.5	**66.3**	**55.2**	74.6	**67.8**	**74.8**

O-CNN pros and cons

+ Performs very well
+ Efficient in memory/compute

- Paper describes only use of basic CNN modules, while latest updates seem to result in much better performance but not well documented

4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks

Christopher Choy
chrischoy@stanford.edu

JunYoung Gwak
jgwak@stanford.edu

Silvio Savarese
ssilvio@stanford.edu

- Minkowski Engine enables convolution with sparse tensors
  - 3D: XYZ
  - 4D: XYZ + time

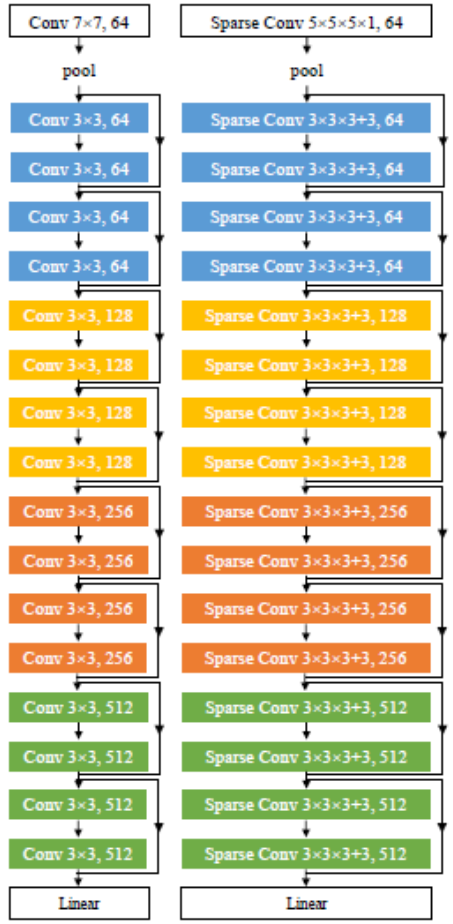


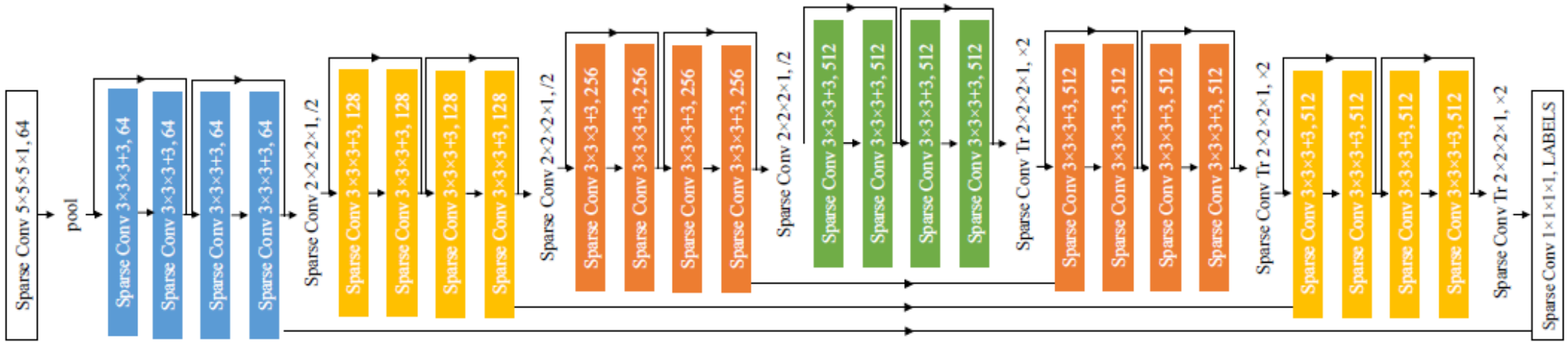
Figure 4: Architecture of ResNet18 (left) and MinkowskiNet18 (right). Note the structural similarity. × indicates a hypercubic kernel, + indicates a hypercross kernel. (best viewed on display)

4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks

Christopher Choy
chrischoy@stanford.edu

JunYoung Gwak
jgwak@stanford.edu

Silvio Savarese
ssilvio@stanford.edu



Architecture of MinkowskiUNet32. $\times$ indicates a hypercubic kernel, $+$ indicates a hypercross kernel. (bes

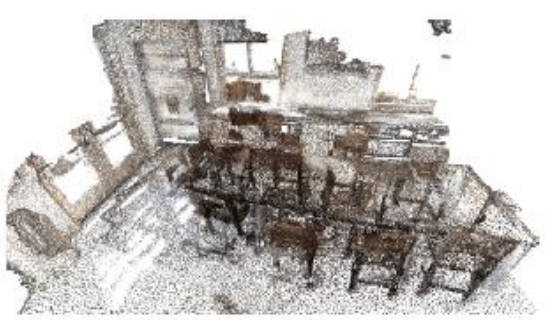
ScanNet / Stanford Dataset

- Process entire room fully convolutionally

RGB

Pred

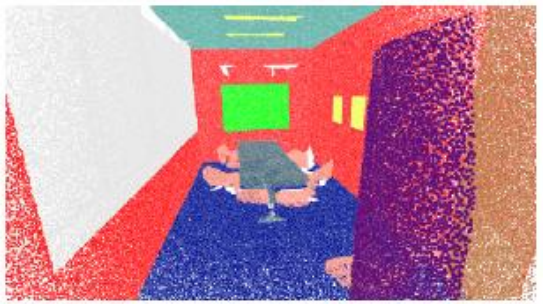
GT

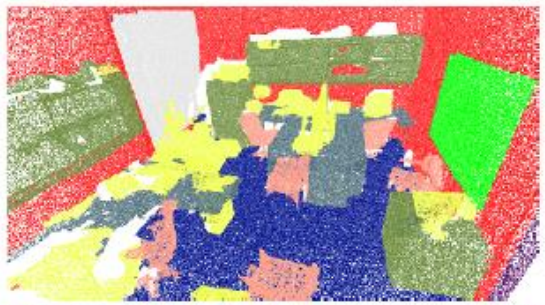













- Good performance due to ability for high resolution voxelization and deep networks

Table 1: 3D Semantic Label Benchmark on ScanNet† [5]

Method	mIOU
ScanNet [5]	30.6
SSC-UNet [10]	30.8
PointNet++ [23]	33.9
ScanNet-FTSDF	38.3
SPLATNet [28]	39.3
TangetConv [29]	43.8
SurfaceConv [20]	44.2
3DMV‡ [6]	48.4
3DMV-FTSDF‡	50.1
PointNet++SW	52.3
MinkowskiNet42 (5cm)	**67.9**
SparseConvNet [10]†	72.5
MinkowskiNet42 (2cm)†	**73.4**

†: post-CVPR submissions. ‡: uses 2D images additionally. Per class IoU in the supplementary material. The parenthesis next to our methods indicate the voxel size.

Table 4: Stanford Area 5 Test (Fold #1) (S3DIS) [2]

Method	mIOU	mAcc
PointNet [22]	41.09	48.98
SparseUNet [9]	41.72	64.62
SegCloud [30]	48.92	57.35
TangentConv [29]	52.8	60.7
3D RNN [32]	53.4	71.3
PointCNN [15]	57.26	63.86
SuperpointGraph [14]	58.04	66.5
MinkowskiNet20	62.60	69.62
MinkowskiNet32	**65.35**	**71.71**

Per class IoU in the supplementary material.

Table 6: Time (s) to process 3D videos with 3D and 4D MinkNet, the volume of a scan at each time step is 50m× 50m × 50m

Voxel Size	0.6m			0.45m			0.3m		
Video Length (s)	3D	4D	4D-CRF	3D	4D	4D-CRF	3D	4D	4D-CRF
3	0.18	0.14	0.17	0.25	0.22	0.27	0.43	0.49	0.59
5	0.31	0.23	0.27	0.41	0.39	0.47	0.71	0.94	1.13
7	0.43	0.31	0.38	0.58	0.61	0.74	0.99	1.59	2.02

Minkowski pros and cons

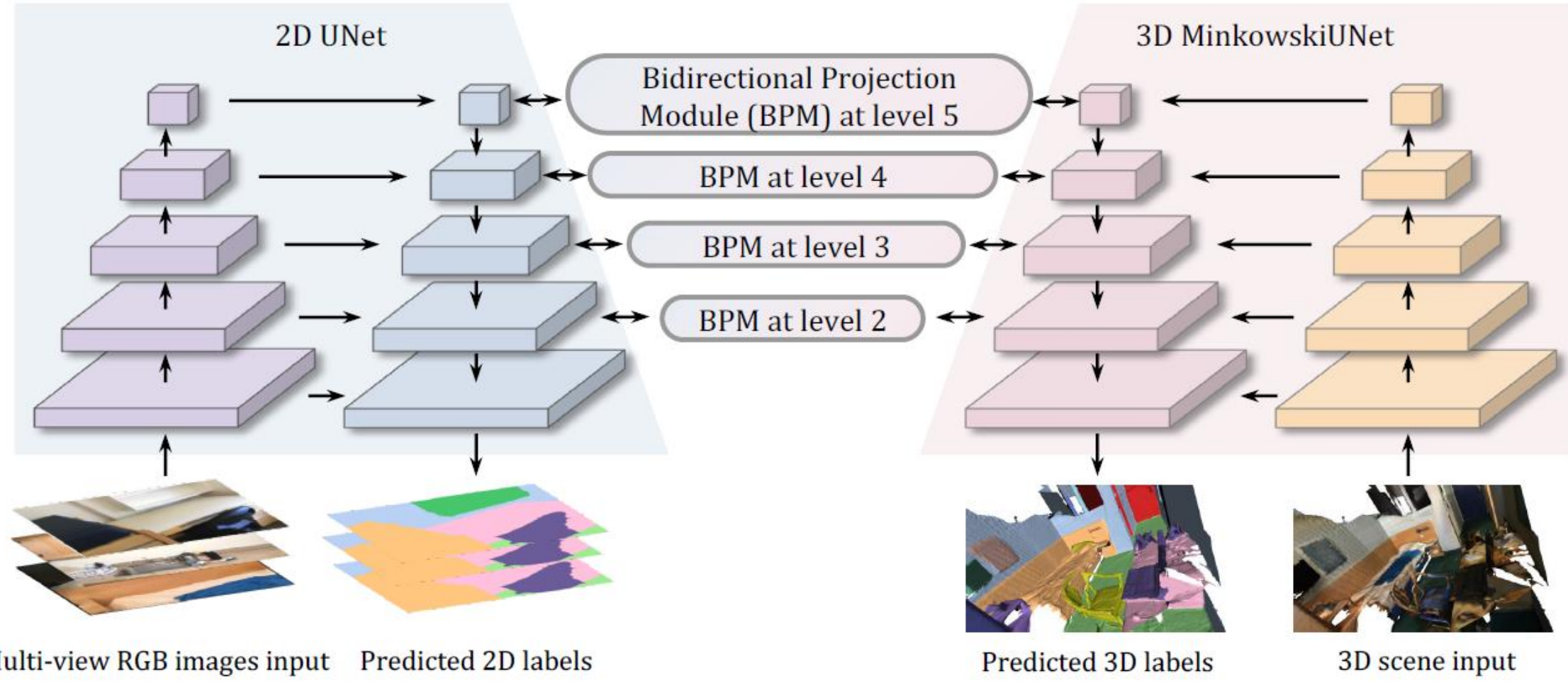
+ Framework for sparse convolution

+ Good accuracy, due to ability for deeper networks

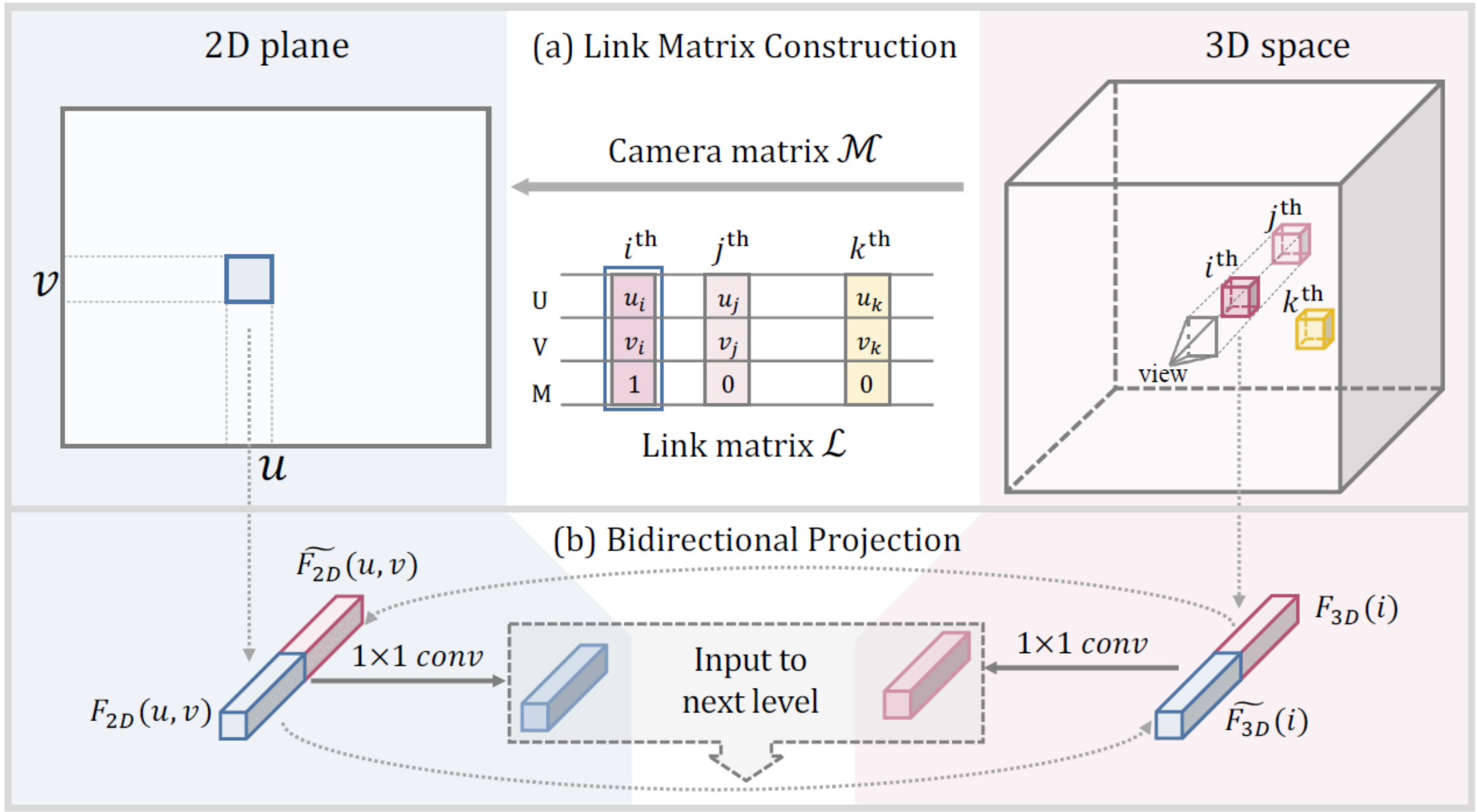
- Does not cite O-CNN, so it’s hard to tell how they compare

Bidirectional Projection Network for Cross Dimension Scene Understanding

Wenbo Hu[1,3*] Hengshuang Zhao[2*] Li Jiang[1] Jiaya Jia[1] Tien-Tsin Wong[1,3†]
[1]The Chinese University of Hong Kong [2]University of Oxford
[3]Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, SIAT, CAS



Bidirectional projection

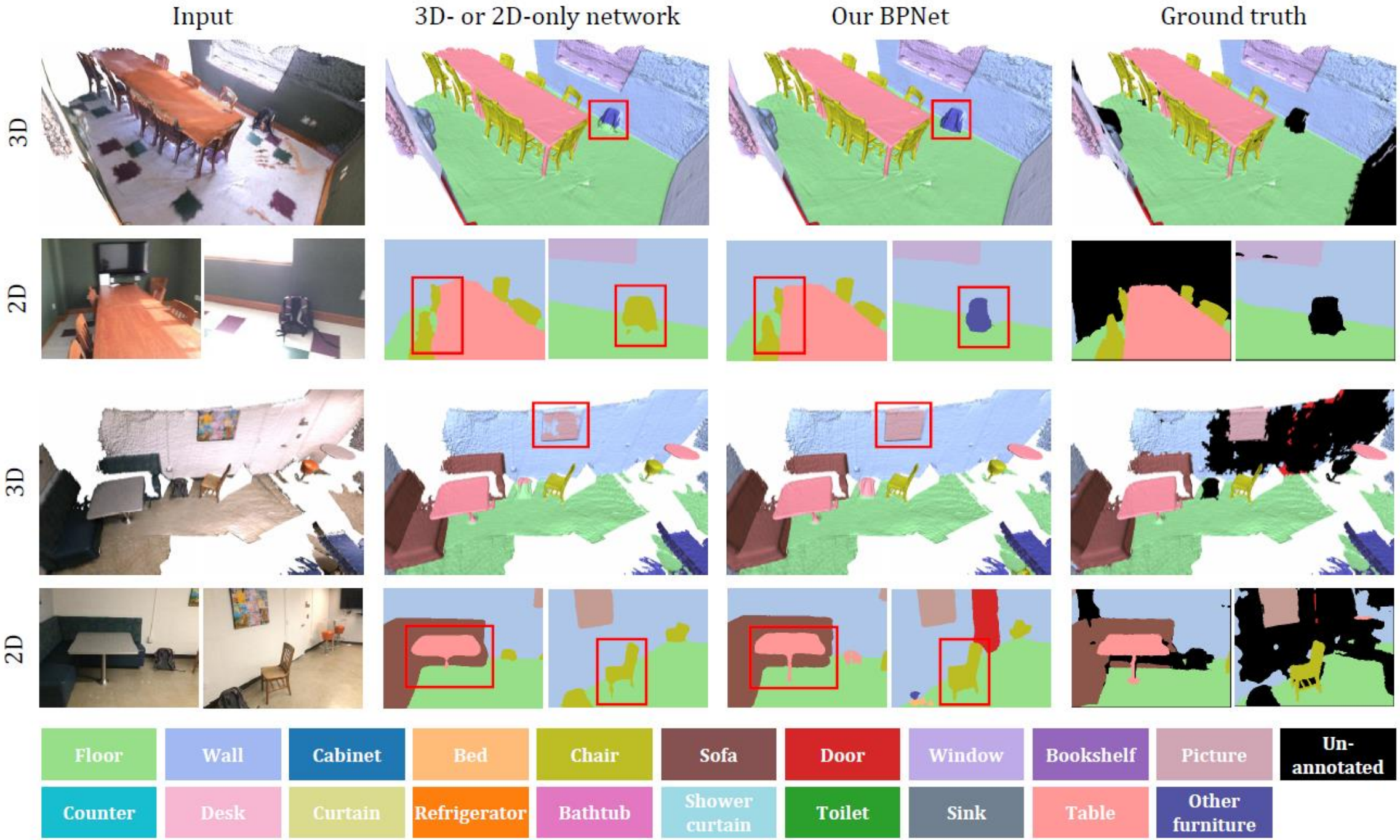


3D ScanNet

Method	mIoU	bath	bed	bkshf	cab	chair	cntr	curt	desk	door	floor	other	pic	fridge	shower	sink	sofa	table	toilet	wall	window
PointNet++ [43]	33.9	58.4	47.8	45.8	25.6	36.0	25.0	24.7	27.8	26.1	67.7	18.3	11.7	21.2	14.5	36.4	34.6	23.2	54.8	52.3	25.2
SPLATNet† [50]	39.3	47.2	51.1	60.6	31.1	65.6	24.5	40.5	32.8	19.7	92.7	22.7	00.0	00.1	24.9	27.1	51.0	38.3	59.3	69.9	26.7
3DMV† [8]	48.4	48.4	53.8	64.3	42.4	60.6	31.0	57.4	43.3	37.8	79.6	30.1	21.4	53.7	20.8	47.2	50.7	41.3	69.3	60.2	53.9
FAConv[69]	63.0	60.4	74.1	76.6	59.0	74.7	50.1	73.4	50.3	52.7	91.9	45.4	32.3	55.0	42.0	67.8	68.8	54.4	89.6	79.5	62.7
MCCNN [19]	63.3	86.6	73.1	77.1	57.6	80.9	41.0	68.4	49.7	49.1	94.9	46.6	10.5	58.1	64.6	62.0	68.0	54.2	81.7	79.5	61.8
FPConv [33]	63.9	78.5	76.0	71.3	60.3	79.8	39.2	53.4	60.3	52.4	94.8	45.7	25.0	53.8	72.3	59.8	69.6	61.4	87.2	79.9	56.7
MVPNet† [24]	64.1	83.1	71.5	67.1	59.0	78.1	39.4	67.9	64.2	55.3	93.7	46.2	25.6	64.9	40.6	62.6	69.1	66.6	87.7	79.2	60.8
DCM-Net [47]	65.8	77.8	70.2	80.6	61.9	81.3	46.8	69.3	49.4	52.4	94.1	44.9	29.8	51.0	82.1	67.5	72.7	56.8	82.6	80.3	63.7
PointConv [62]	66.6	78.1	75.9	69.9	64.4	82.2	47.5	77.9	56.4	50.4	95.3	42.8	20.3	58.6	75.4	66.1	75.3	58.8	90.2	81.3	64.2
PointASNL [64]	66.6	70.3	78.1	75.1	65.5	83.0	47.1	76.9	47.4	53.7	95.1	47.5	27.9	63.5	69.8	67.5	75.1	55.3	81.6	80.6	70.3
KP-FCNN [55]	68.4	84.7	75.8	78.4	64.7	81.4	47.3	77.2	60.5	59.4	93.5	45.0	18.1	58.7	80.5	69.0	78.5	61.4	88.2	81.9	63.2
MinkowskiNet [6]	73.6	85.9	81.8	83.2	70.9	84.0	52.1	85.3	66.0	64.3	95.1	54.4	28.6	73.1	89.3	67.5	77.2	68.3	87.4	85.2	72.7
BPNet (Ours)†	74.9	90.9	81.8	81.1	75.2	83.9	48.5	84.2	67.3	64.4	95.7	52.8	30.5	77.3	85.9	78.8	81.8	69.3	91.6	85.6	72.3

Table 1. Comparison with the typical streams of methods on ScanNetV2 3D Semantic label benchmark, including point cloud based, sparse convolution based, and joint 2D-3D-input (marked with †) based methods.

3 views performs best


Input
3D- or 2D-only network
Our BPNet
Ground truth
3D
2D
3D
2D
Floor
Wall
Cabinet
Bed
Chair
Sofa
Door
Window
Bookshelf
Picture
Un-annotated
Counter
Desk
Curtain
Refrigerator
Bathtub
Shower curtain
Toilet
Sink
Table
Other furniture

ScanNet 2D

Method	mIoU	bath	bed	bkshf	cab	chair	cntr	curt	desk	door	floor	other	pic	fridge	shower	sink	sofa	table	toilet	wall	window
PSPNet [72]	47.5	49.0	58.1	28.9	50.7	6.7	37.9	61.0	41.7	43.5	82.2	27.8	26.7	50.3	22.8	61.6	53.3	37.5	82.0	72.9	56.0
UNet34 [46]	48.9	55.3	62.6	26.6	50.3	23.5	37.9	52.4	49.8	41.6	84.5	28.6	32.1	54.0	12.8	60.8	55.3	38.5	81.6	73.6	56.6
3DMV [8]	49.8	48.1	61.2	57.9	45.6	34.3	38.4	62.3	52.5	38.1	84.5	25.4	26.4	55.7	18.2	58.1	59.8	42.9	76.0	66.1	44.6
FuseNet† [16]	53.5	57.0	68.1	18.2	51.2	29.0	43.1	65.9	50.4	49.5	90.3	30.8	42.8	52.3	36.5	67.6	62.1	47.0	76.2	77.9	54.1
SSMA† [56]	57.7	69.5	71.6	43.9	56.3	31.4	44.4	71.9	55.1	50.3	88.7	34.6	34.8	60.3	35.3	70.9	60.0	45.7	90.1	78.6	59.9
RFBNet† [10]	59.2	61.6	75.8	65.9	58.1	33.0	46.9	65.5	54.3	52.4	92.4	35.5	33.6	57.2	47.9	67.1	64.8	48.0	81.4	81.4	61.4
BPNet (Ours)†	67.0	82.2	79.5	83.6	65.9	48.1	45.1	76.9	65.6	56.7	93.1	39.5	39.0	70.0	53.4	68.9	77.0	57.4	86.5	83.1	67.5

(mark indicates 2d-3d)

BPNet pros and cons

+ Incorporates both image and 3D features in an elegant way

+ Performs very well

- Must be slow (?) -- might not be worth complexity vs Minkowski or O-CNN

Open problems / research ideas

- MVS point clouds
  - Most or all datasets are currently based on laser scans or other detailed depth sensors
  - MVS is challenging due to noisy and incomplete points
- Automated progress monitoring
  - Detect presence/state of building elements given point clouds and images
  - Challenges of many element types, long-tail distribution
- Change/deviation detection
  - Given two 3D models, identify the important differences, e.g. deviation from design or change detection from date to date

Summary

- Early successful approaches (PointNet variety) focus on point-wise processing, or graph-based approaches in local neighborhoods
- Current best-performing approaches (O-CNN, Minkowski) use sparse convolution
- Semantic segmentation on MVS point clouds is relatively unstudied and may raise new challenges