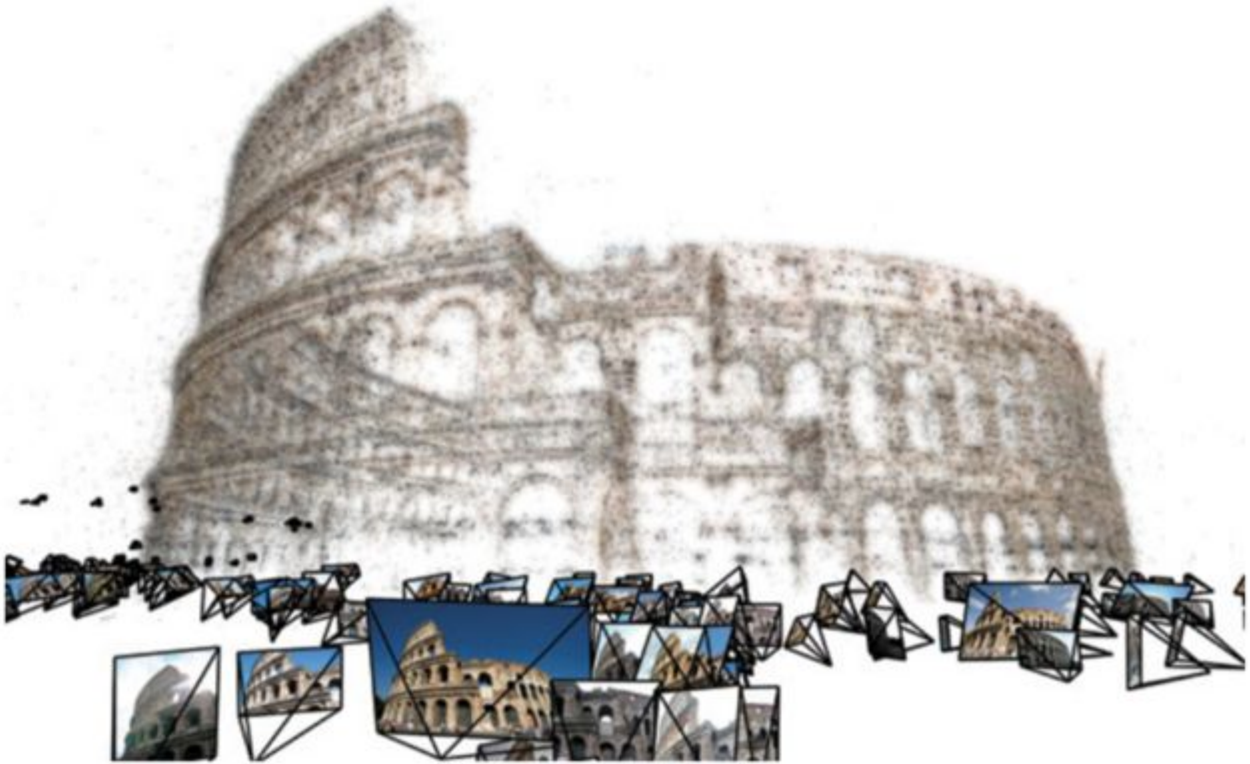
Deep Hough Voting
3D Object Detection in Point Clouds

Or Litany

FAIR / Stanford

In collaboration with: Charles Qi, Kaiming He and Leonidas Guibas

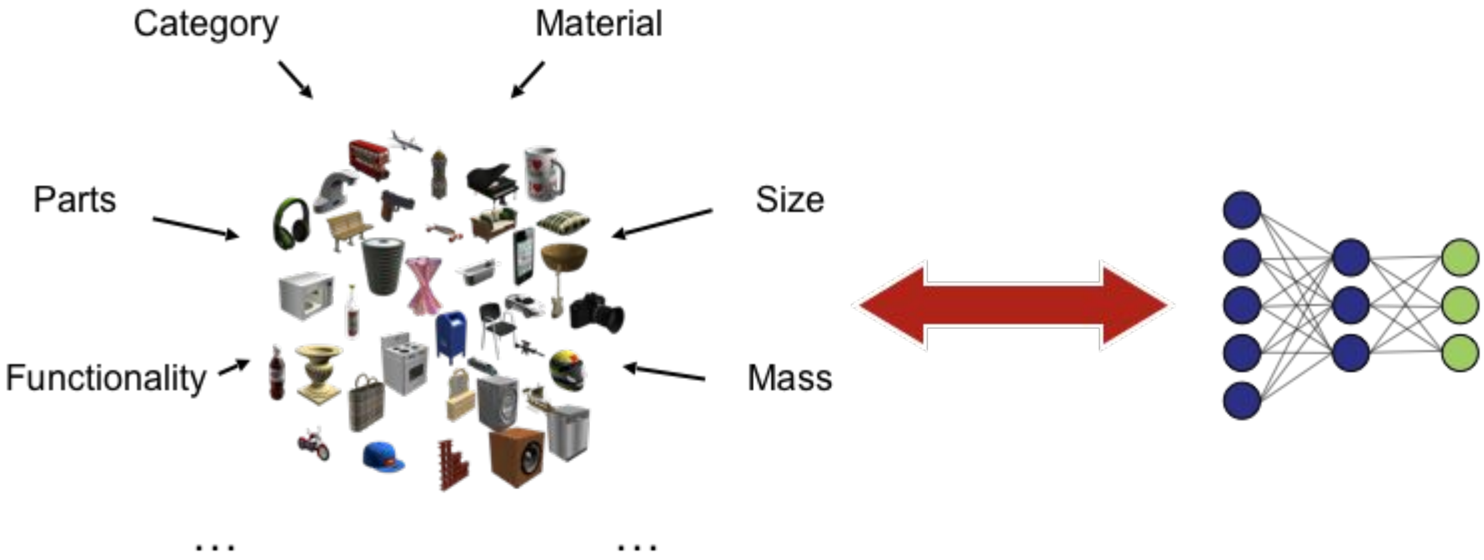
3D is a natural representation of the world



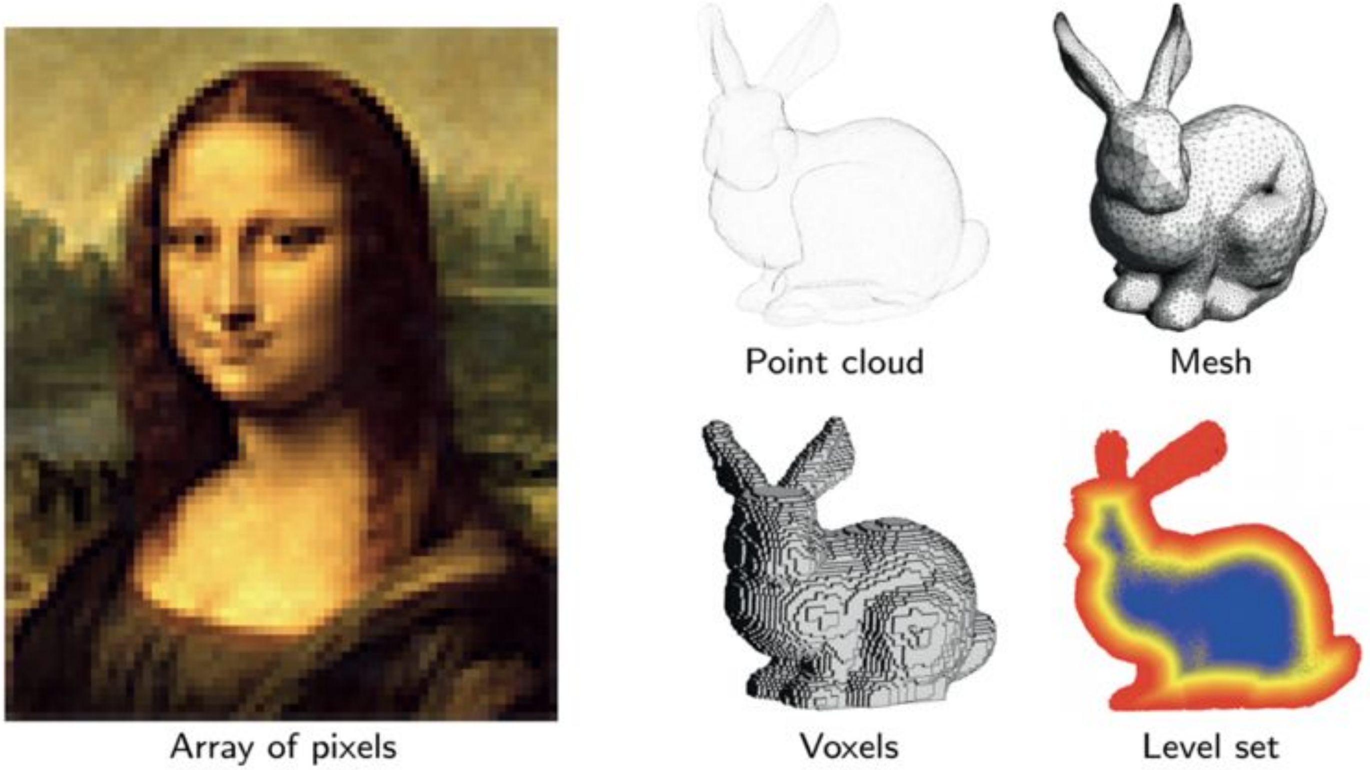
Agarwal et. al., ICCV'09

3D consumer market

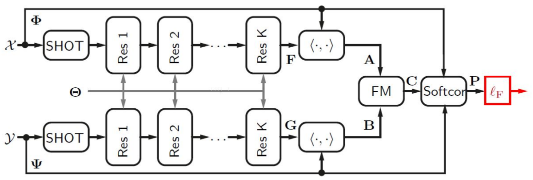
Data driven tools for 3D



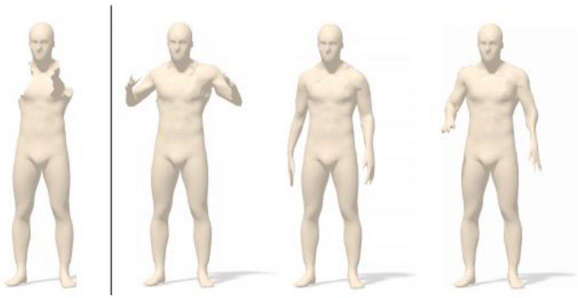
3D representations



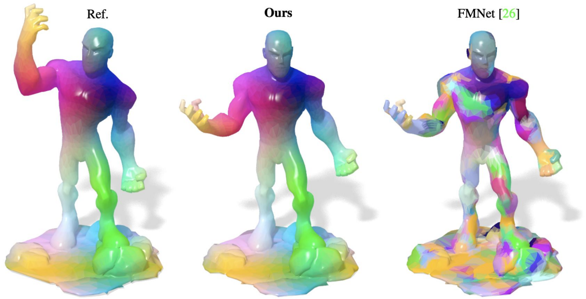
Learning on graphs and manifolds (shameless plug)



FM*Net*, ICCV'17



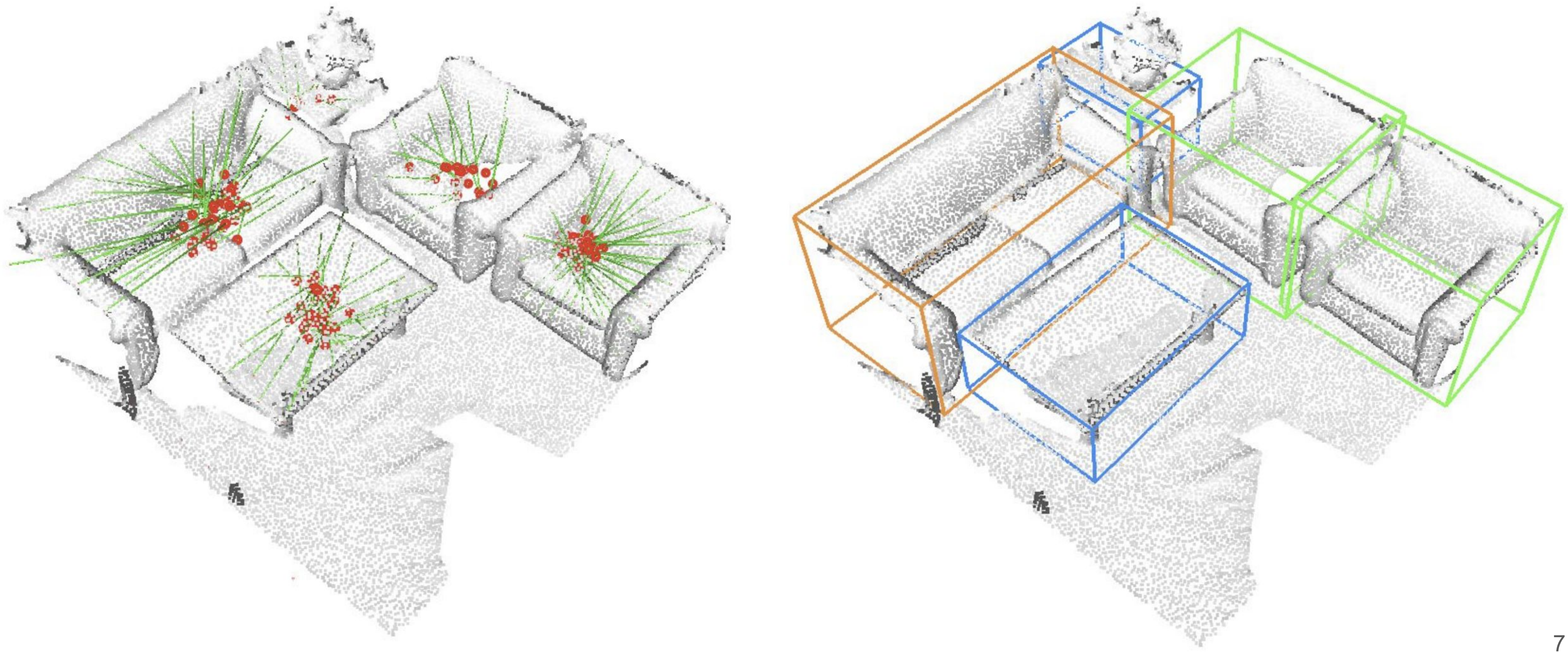
Shape completion, CVPR'18



self-supervised, CVPR'19

What if the graph (connectivity) is unknown?

Deep Hough Voting: 3D Object Detection in Point Clouds



What is 3D object detection?

Generally: To localize and recognize objects in a 3D scene.

What is 3D object detection?

Generally: To localize and recognize objects in a 3D scene.

Specifically in literature: Estimate amodal, oriented 3D bounding boxes and semantic classes of objects from 3D point clouds or RGB-D data.

What is 3D object detection?

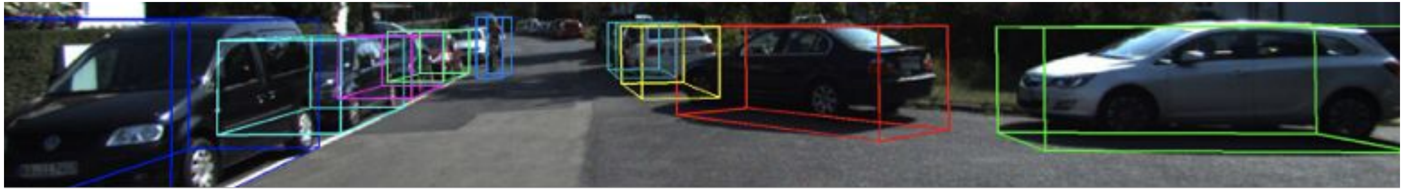
Generally: To localize and recognize objects in a 3D scene.

Specifically in literature: Estimate amodal, oriented 3D bounding boxes and semantic classes of objects from 3D point clouds or RGB-D data.

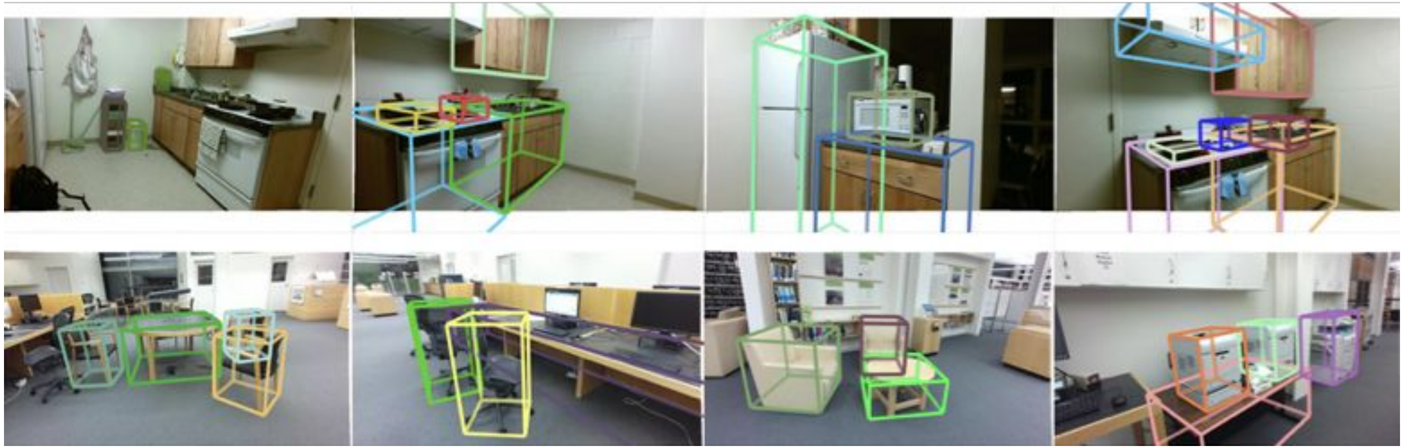
Applications:

- Augmented reality.
- Robotics.
- Autonomous driving.

What is 3D object detection?



KITTI



SUN RGB-D

What is 3D object detection?

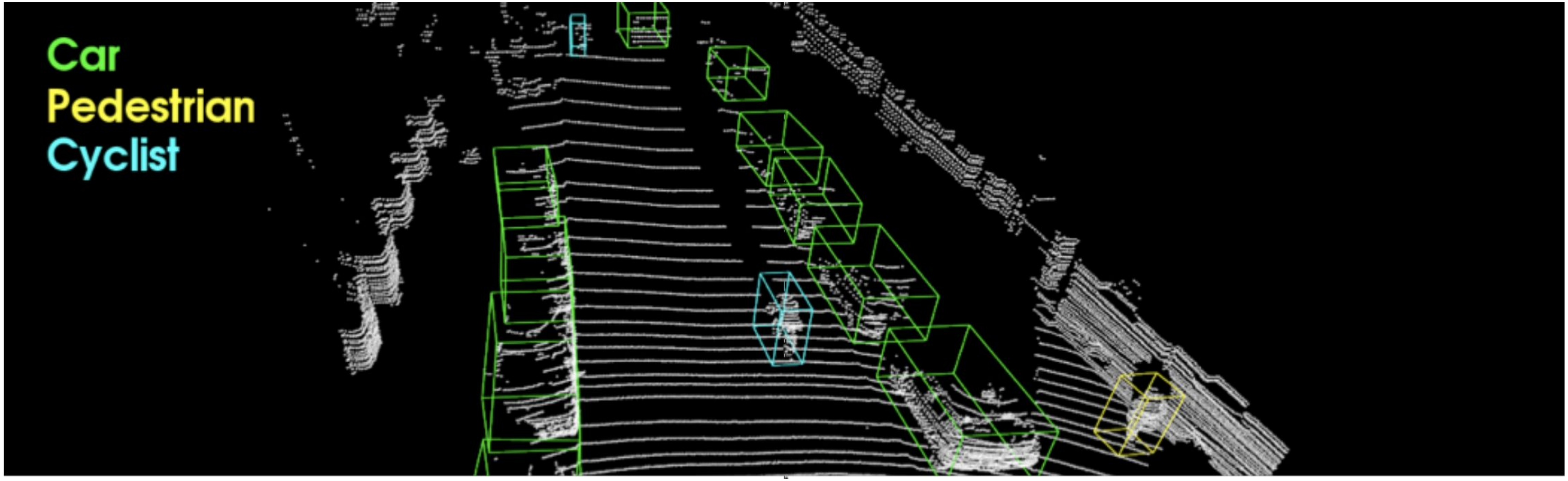
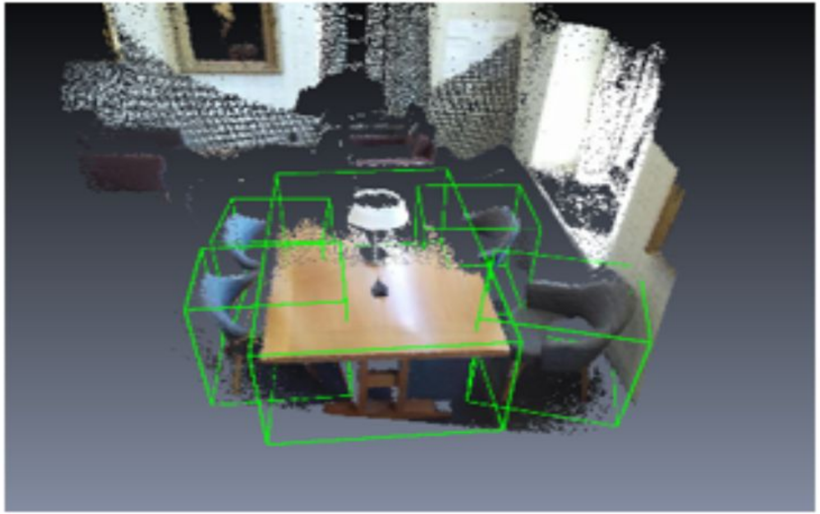


Figure by Yin et al. (VoxelNet)

What is 3D object detection?



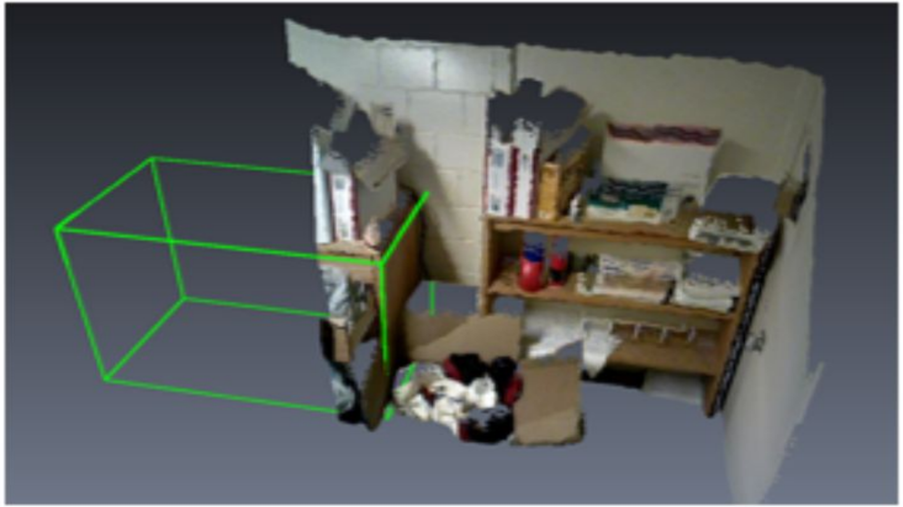
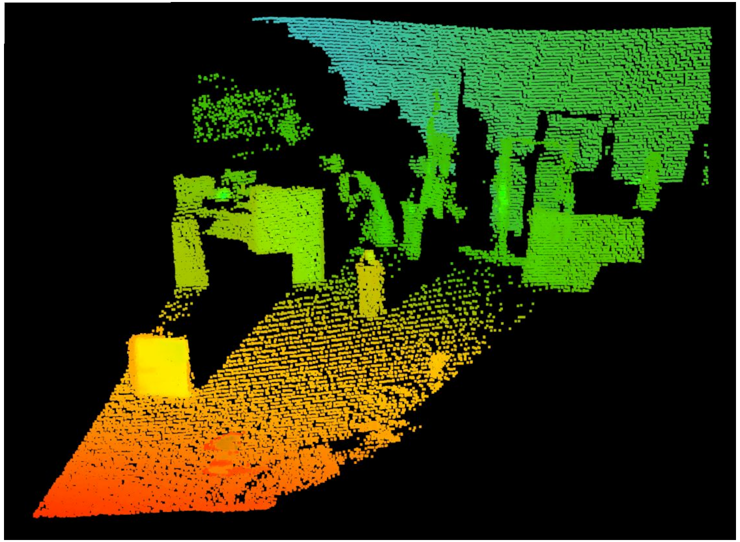


Figure by Lahoud et al. (2D-driven 3D object detection)

3D Vs. 2D Object Detection

3D input: **point clouds** from Lidar, RGB-D, reconstructed meshes.

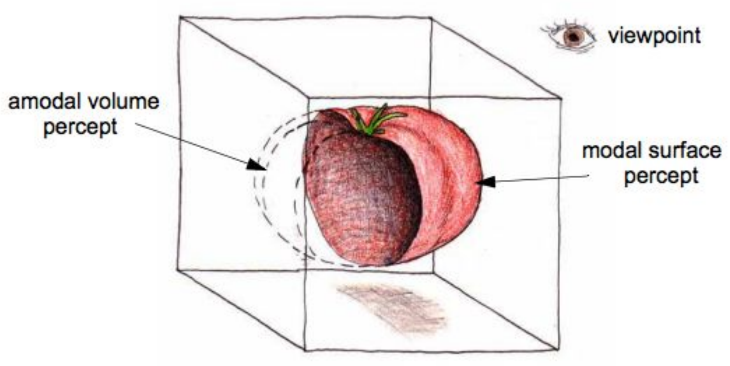


- \+ **Accurate 3D geometry (depth and scale)**
- \+ **Robust to illumination**
- \- **Sparse and irregular (doesn't fit with CNNs).**
- \- **Centroid can be far from surface points.**

3D Vs. 2D Object Detection

3D input: **point clouds** from Lidar, RGB-D, reconstructed meshes.

3D output: **Amodal** 3D oriented bounding boxes with semantic classes



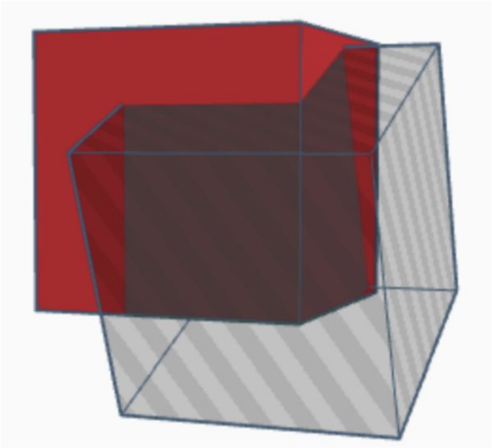
3D box parameterization: $c_x, c_y, c_z \quad h, w, l \quad \theta, \phi, \psi$

Usually we only consider 1D rotation around the up-axis.

Evaluation metric

Average Precision (AP) with a 3D Intersection over Union (IoU) threshold.

90% correct in
each dimension,
perfect angle:
0.9^3 = 0.73!



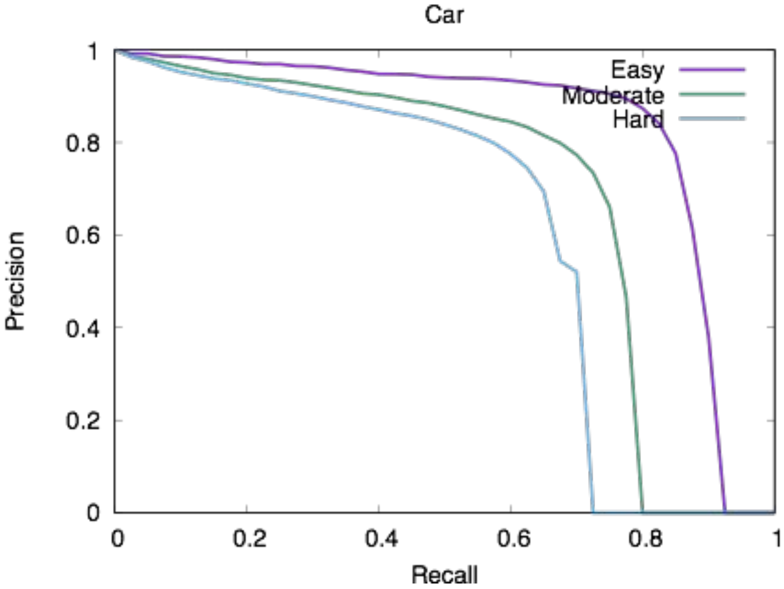


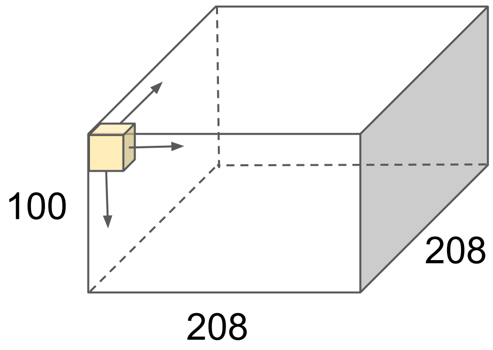
Figure from the SUMO report.

Key research problems

- **3D object proposal** *(challenges: large search space, varying sizes and orientations)*
- How to use image (high resolution, rich semantics, 2D geometry) and 3D (low resolution, accurate 3D geometry)
- How to represent "objects": bounding boxes (2D,3D,oriented,amodal), instance masks, others (convex hulls,voxels,meshes,primitives,…)

3D object proposal: Current methods' limitations

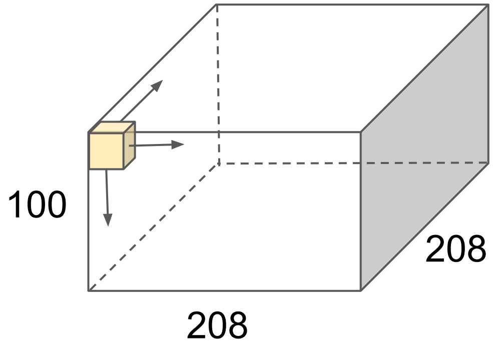
3D CNN detector



- **High computation cost.**
- **Search in empty space (no use of <u>sparsity</u> in point clouds).**

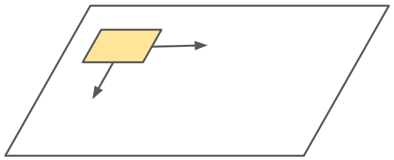
3D object proposal: Current methods’ limitations

3D CNN detector



- **High computation cost.**
- **Search in empty space (no use of <u>sparsity</u> in point clouds).**

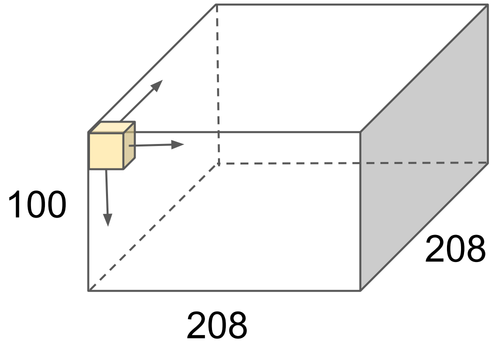
Bird’s eye view detector



- **Restricted to certain types of scenes (e.g. driving).**
- **Essentially a 2D detector.**

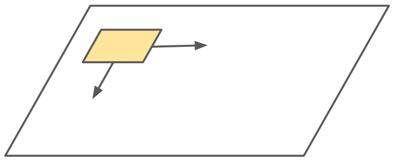
3D object proposal: Current methods' limitations

3D CNN detector



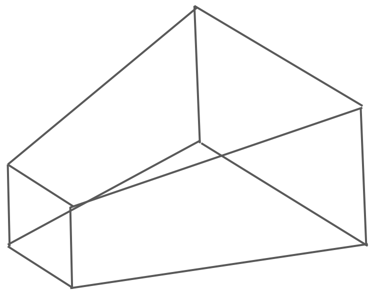
- **High computation cost.**
- **Search in empty space (no use of <u>sparsity</u> in point clouds).**

Bird's eye view detector



- **Restricted to certain types of scenes (e.g. driving).**
- **Essentially a 2D detector.**

Frustum-based detector

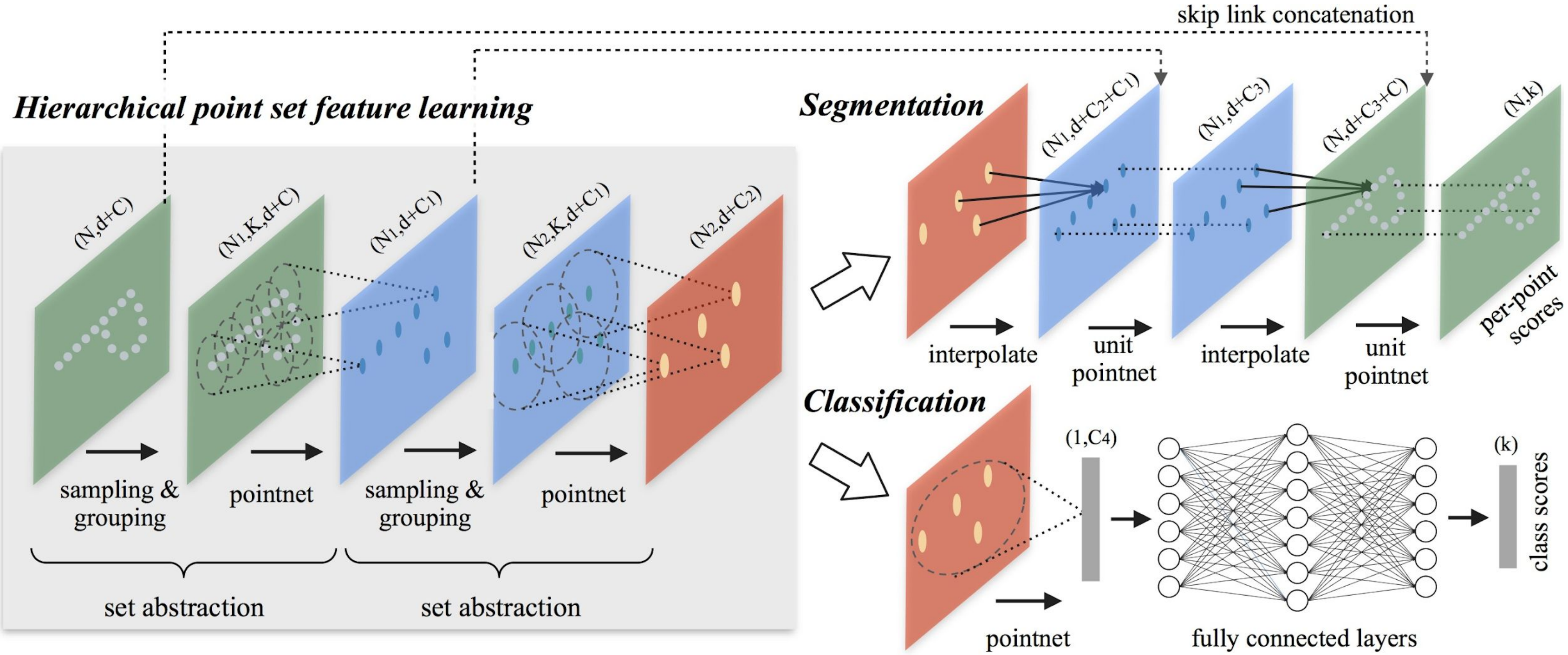


- **Hard dependence on 2D detectors.**

3D object proposal: What we want

- **Generic**: no assumption on canonical viewpoint as in bird's eye view detectors.
- **3D-based**: no hard dependence on 2D images as in frustum-based detectors.
- **Efficient**: no brute-force search in the entire 3D space as in 3D CNNs. Leverage the sparsity in point clouds.

Simple point cloud based solution: Direct prediction



PointNet++, Qi et. al.

Simple point cloud based solution: Direct prediction

- Predict directly from existing points
- **Challenge**: Existing points can be very far from object centers.

3D object proposal:
A return of hough voting!

Hough voting detector recap

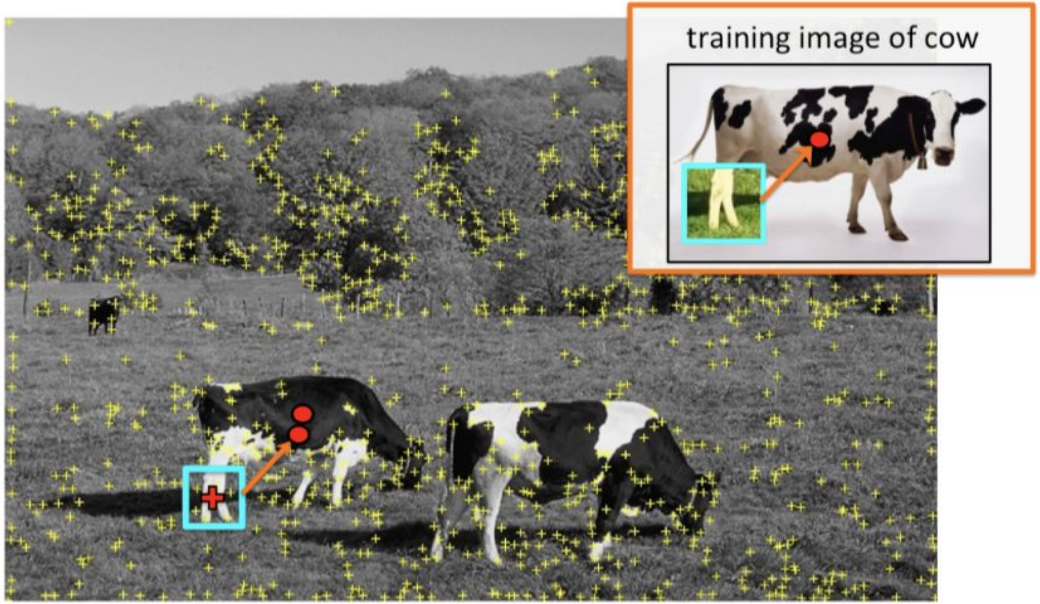
vote for center of object

Hough voting pipeline (in 2D):

- Select interest points
- Match patch around each interest point to a training patch (codebook)
- Vote for object center given that training instance

From U. Toronto CSC420

Hough voting detector recap



vote for center of object

Hough voting pipeline (in 2D):
- Select interest points
- Match patch around each interest point to a training patch (codebook)
- Vote for object center given that training instance

From U. Toronto CSC420

Hough voting detector recap

vote for center of object

Hough voting pipeline (in 2D):
- Select interest points
- Match patch around each interest point to a training patch (codebook)
- Vote for object center given that training instance

From U. Toronto CSC420

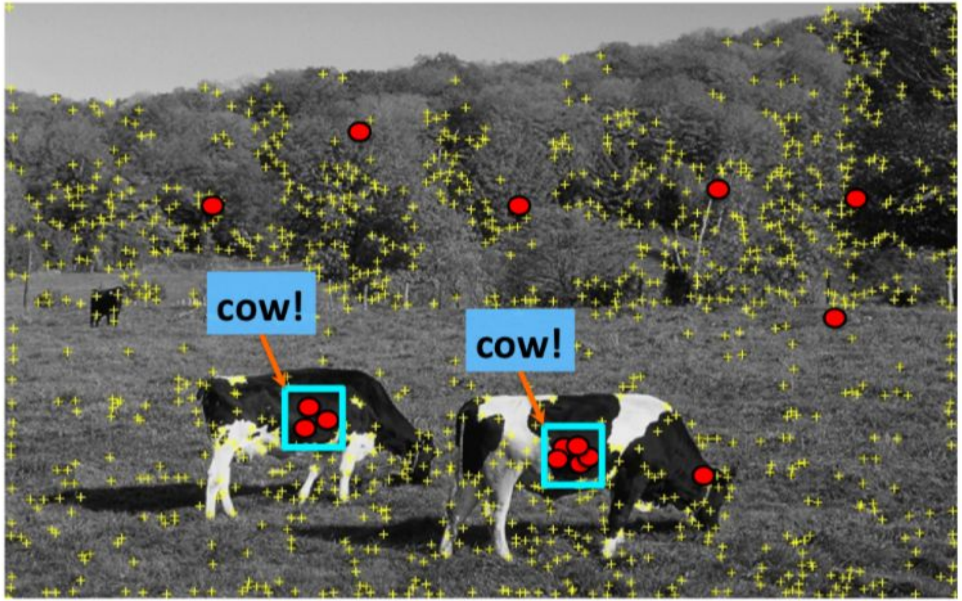
Hough voting detector recap

of course some wrong votes are bound to happen...

Hough voting pipeline (in 2D):
- Select interest points
- Match patch around each interest point to a training patch (codebook)
- Vote for object center given that training instance

From U. Toronto CSC420

Hough voting detector recap



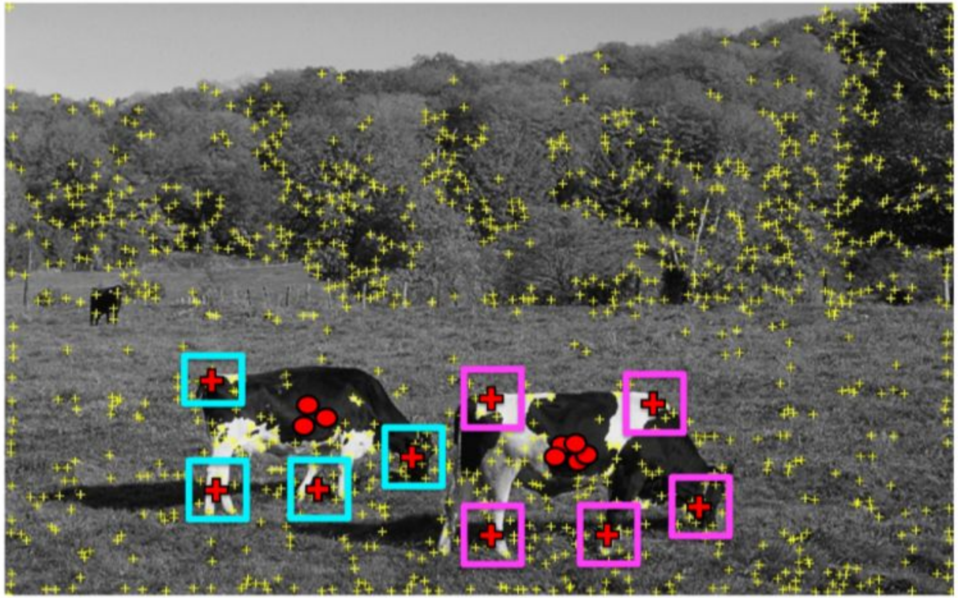
But that's ok. We want only **peaks** in voting space.

Hough voting pipeline (in 2D):

- Select interest points
- Match patch around each interest point to a training patch (codebook)
- Vote for object center given that training instance
- **Votes clustering to find peaks**

From U. Toronto CSC420

Hough voting detector recap



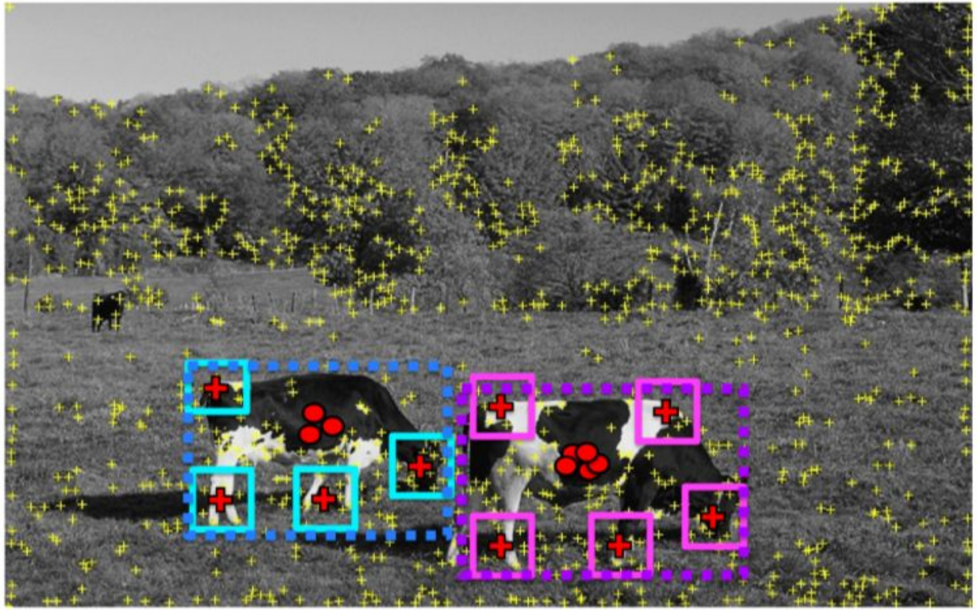
Find patches that voted for the peaks (back-projection).

Hough voting pipeline (in 2D):

- Select interest points
- Match patch around each interest point to a training patch (codebook)
- Vote for object center given that training instance
- Votes clustering to find peaks
- **Find patches that voted for the peaks back-projection**

From U. Toronto CSC420

Hough voting detector recap



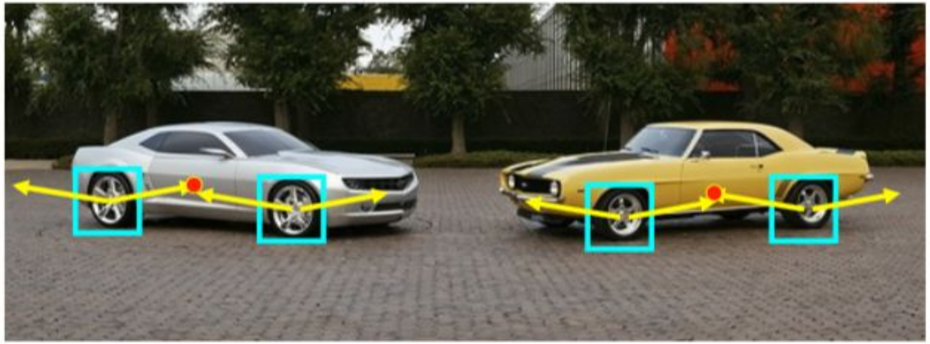
Find full objects based on the back-projected patches.

Hough voting pipeline (in 2D):

- Select interest points
- Match patch around each interest point to a training patch (codebook)
- Vote for object center given that training instance
- Votes clustering to find peaks
- Find patches that voted for the peaks back-projection
- **Find full objects based on back-projected patches**

From U. Toronto CSC420

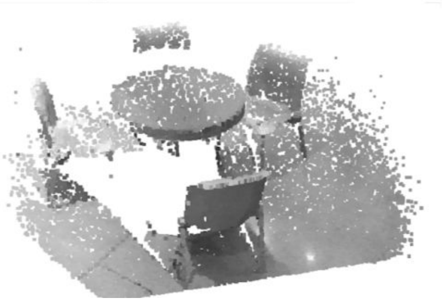
Hough voting detector recap



+ Suitable for sparse data: computation is only on “interest” points
+ Long-range and non-uniform context aggregation
- **Not end-to-end optimizable**

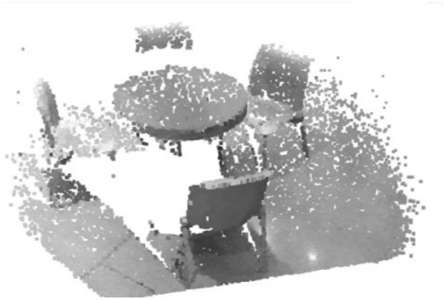
Deep Hough voting:

Input:
point cloud

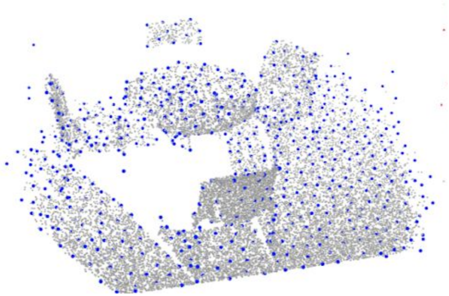


Deep Hough voting:

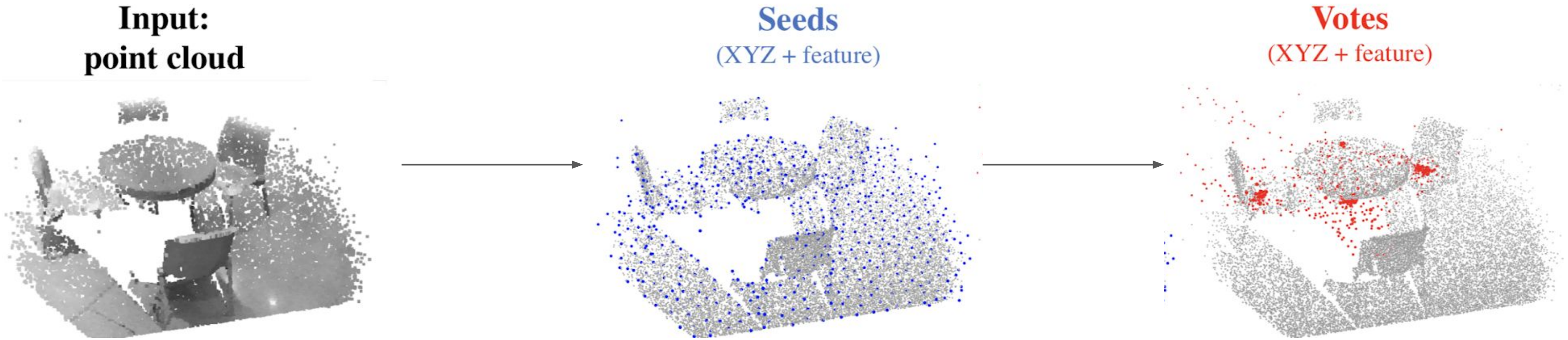
Input:
point cloud



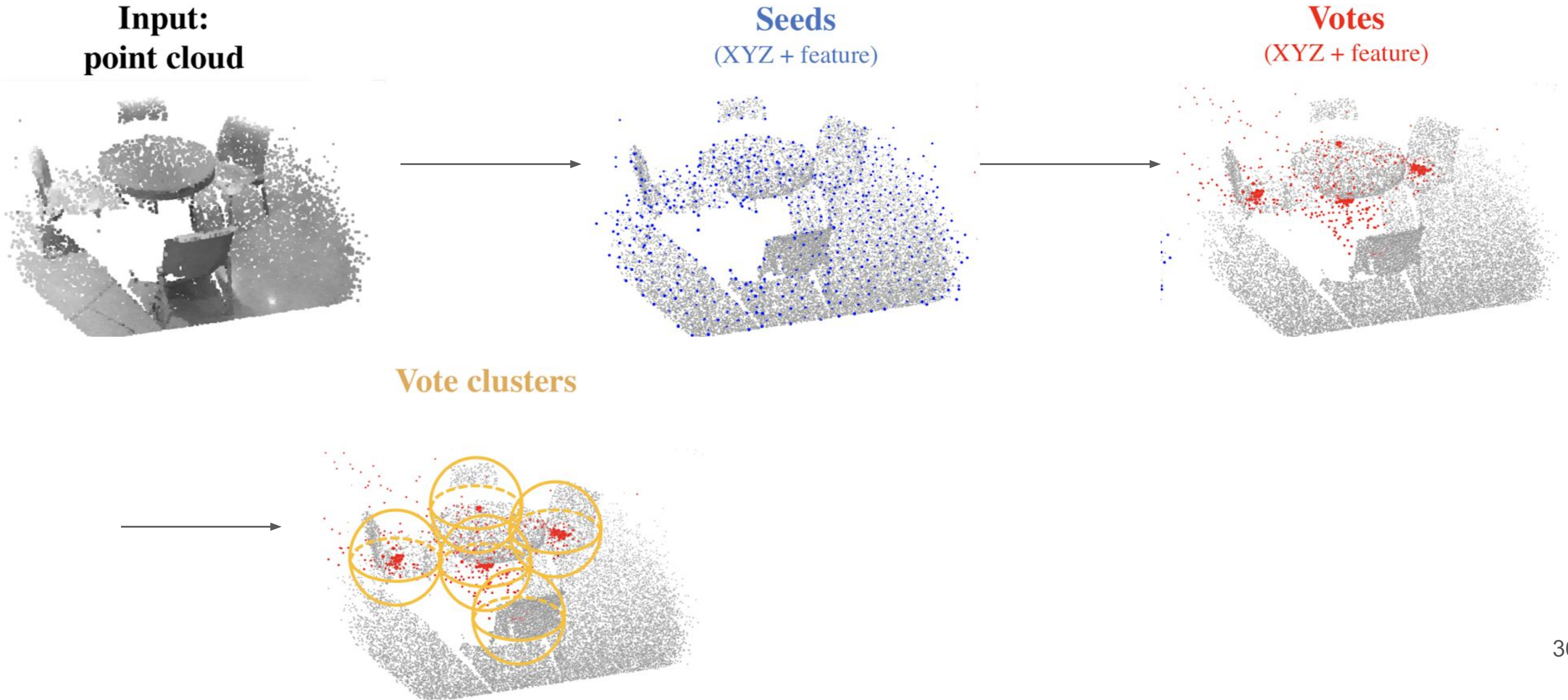
Seeds
(XYZ + feature)



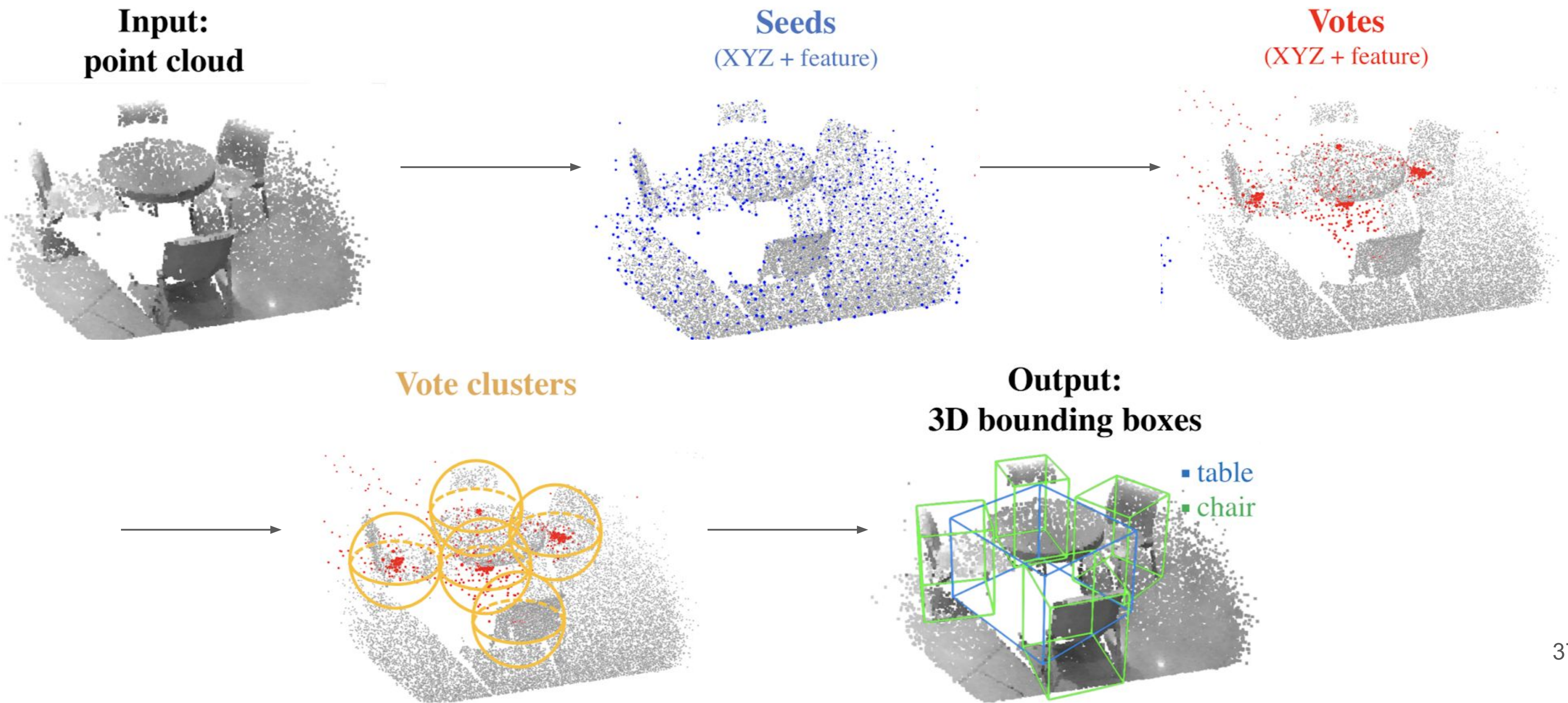
Deep Hough voting:

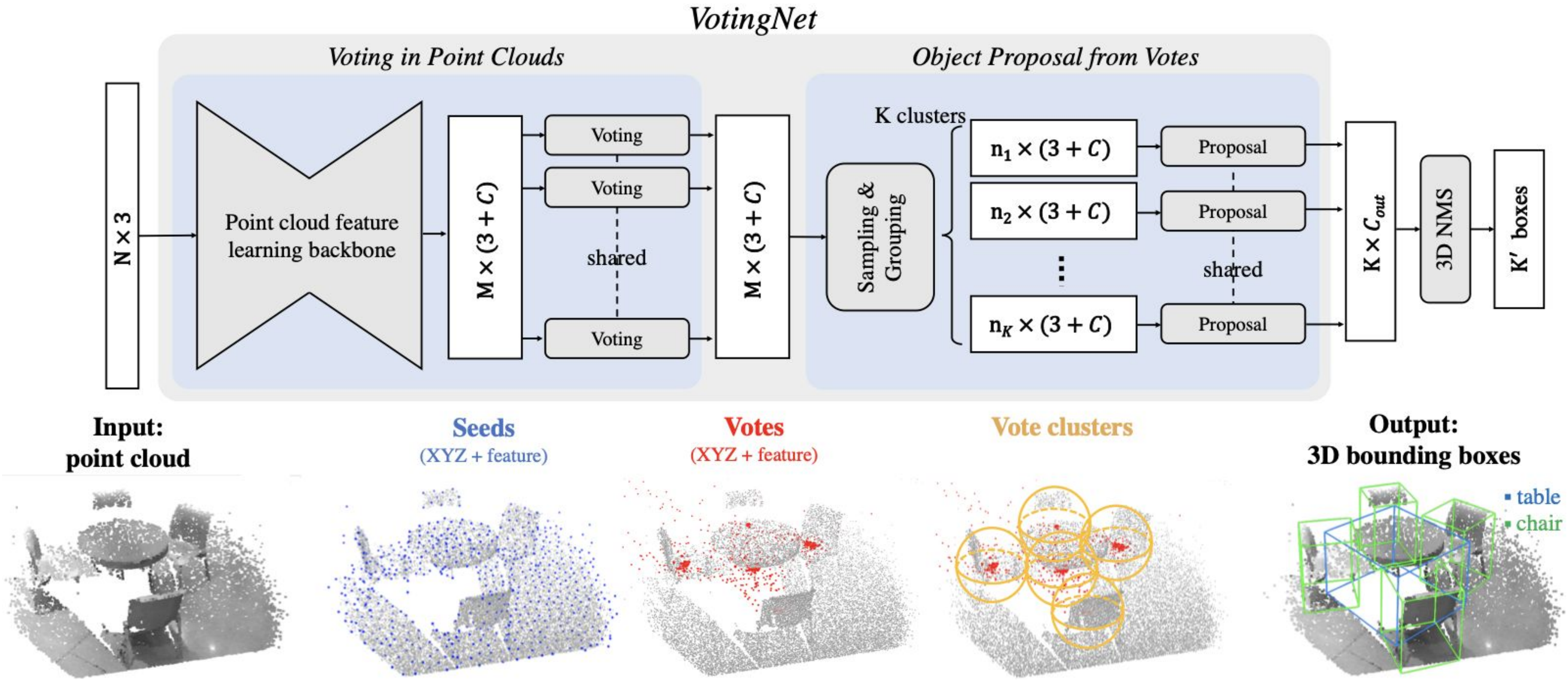


Deep Hough voting:



Deep Hough voting:

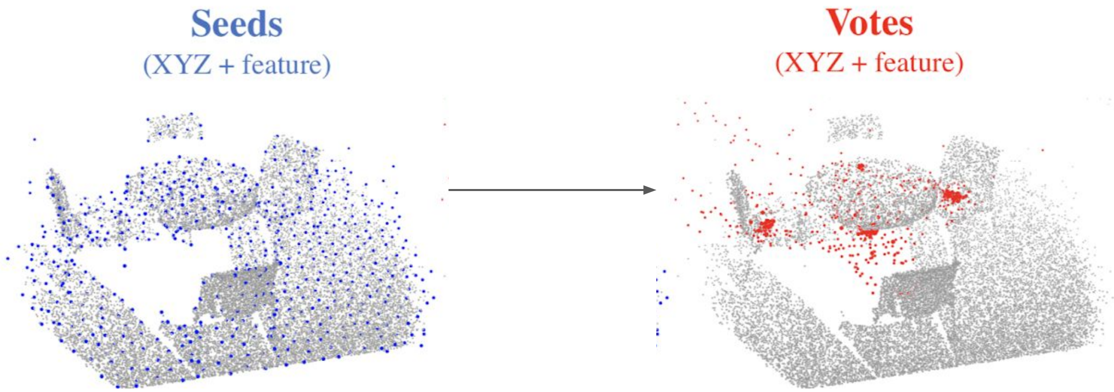




$$L_{\text{VoteNet}} = L_{\text{vote-reg}} + \lambda_1 L_{\text{obj-cls}} + \lambda_2 L_{\text{box}} + \lambda_3 L_{\text{sem-cls}}$$

Deep Hough voting:

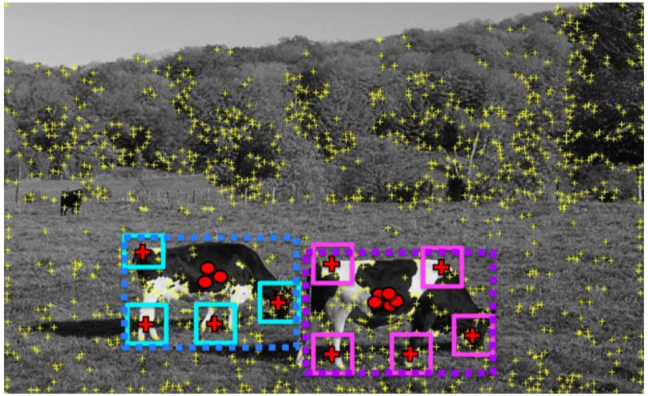
- Votes are "virtual points": same structure, better location



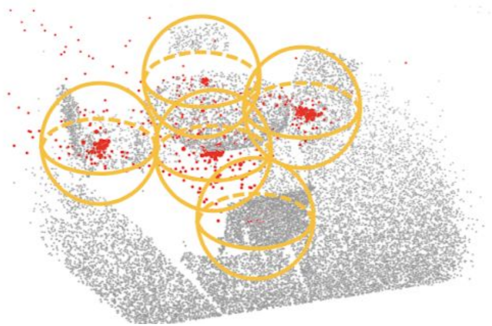
Deep Hough voting:

- Votes are "virtual points": same structure, better location
- Aggregation instead of back-tracing:
  - Learn to filter
  - Predict more than just location: pose, class, etc.
  - Amodal proposals

Back-trace



Learn to aggregate

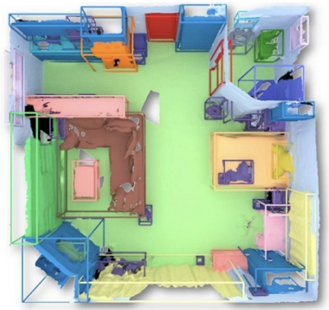


Results

SUN RGB-D

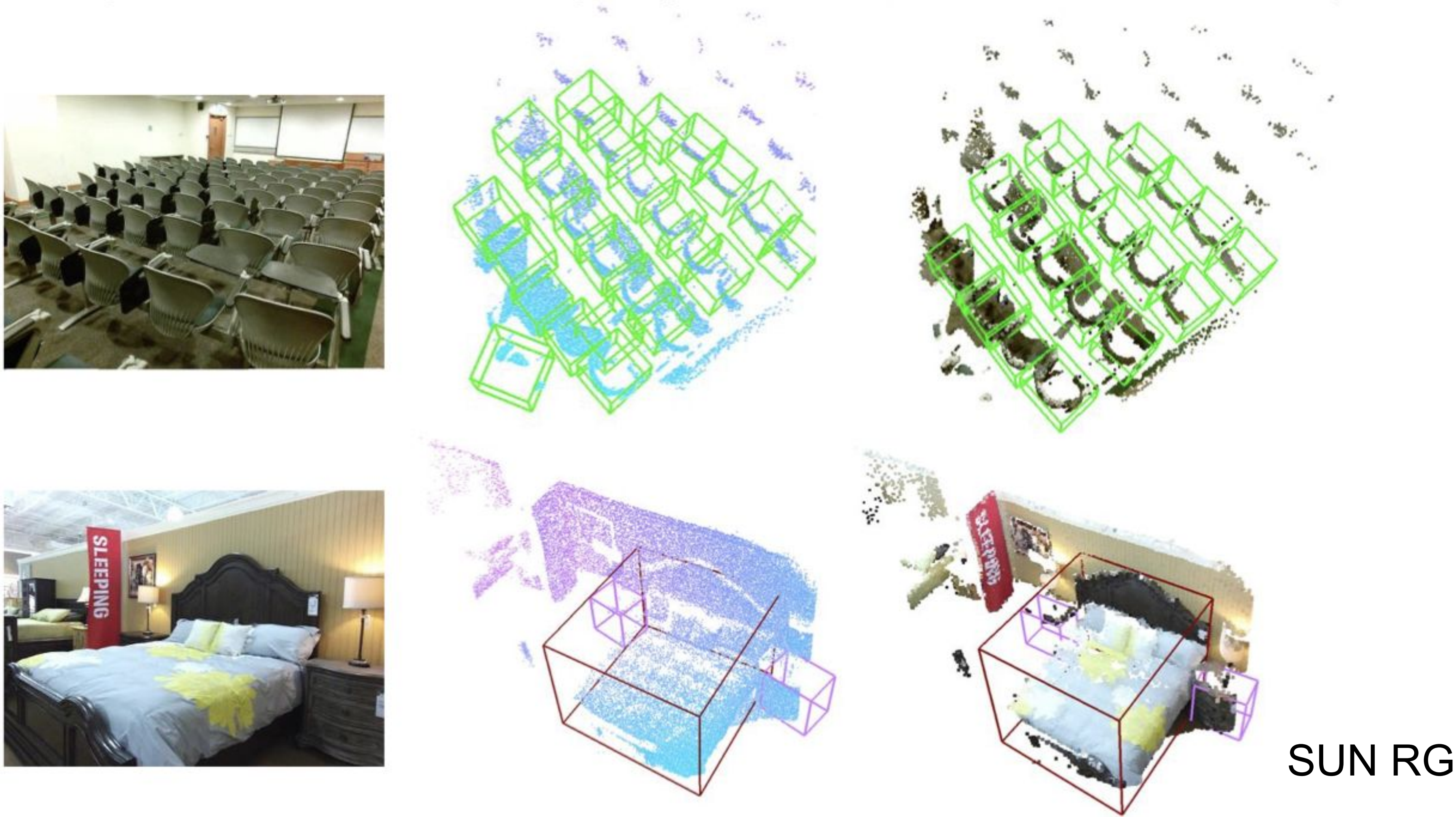
Single RGB-D images
Eval on 10 classes.
5k/5k train/test.
amodal

ScanNet



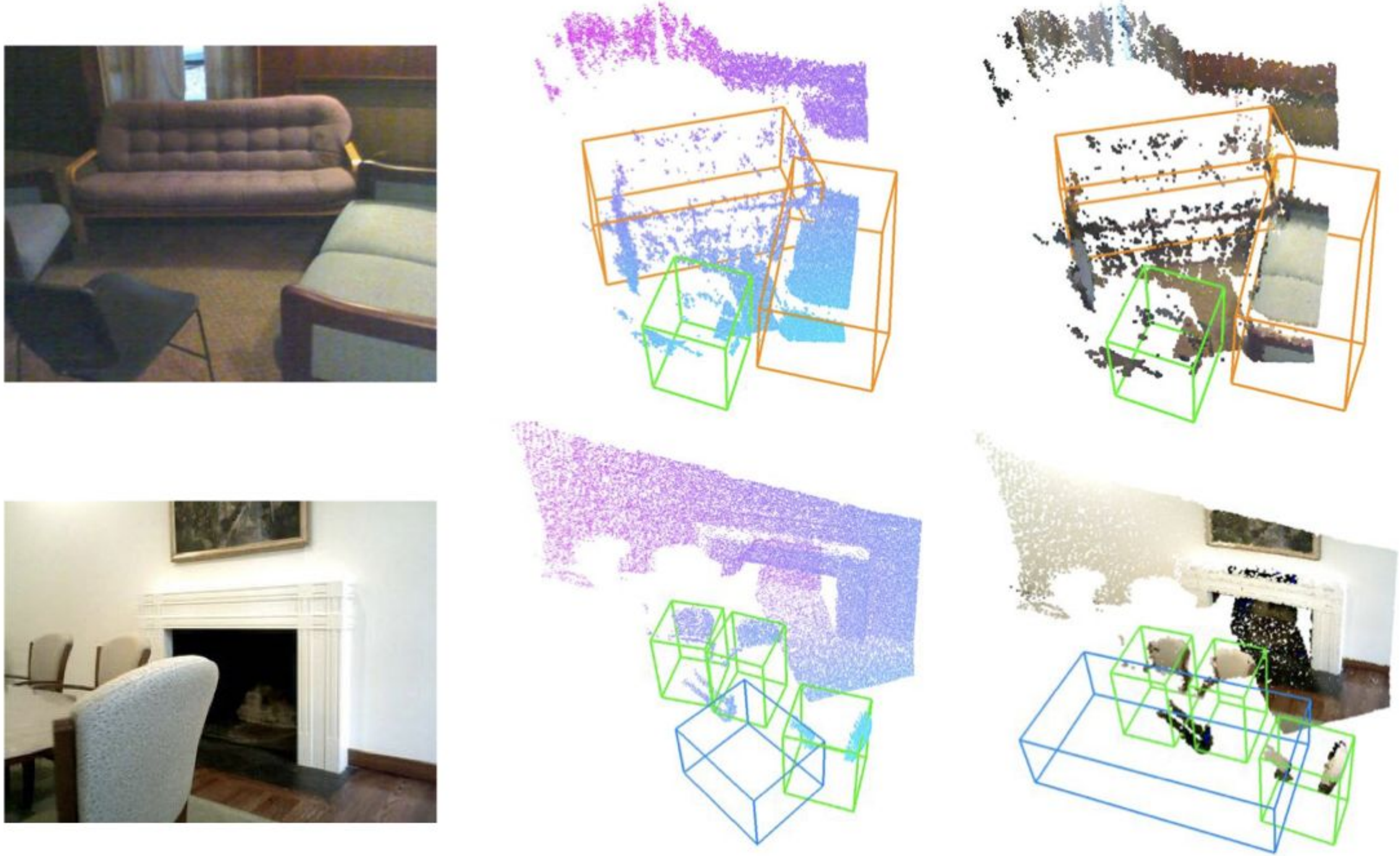
Reconstructed scenes.
Eval on 18 classes.
1.2k/302
train/val
Not amodal, no pose.

Image of the scene | VotingNet prediction | Ground truth



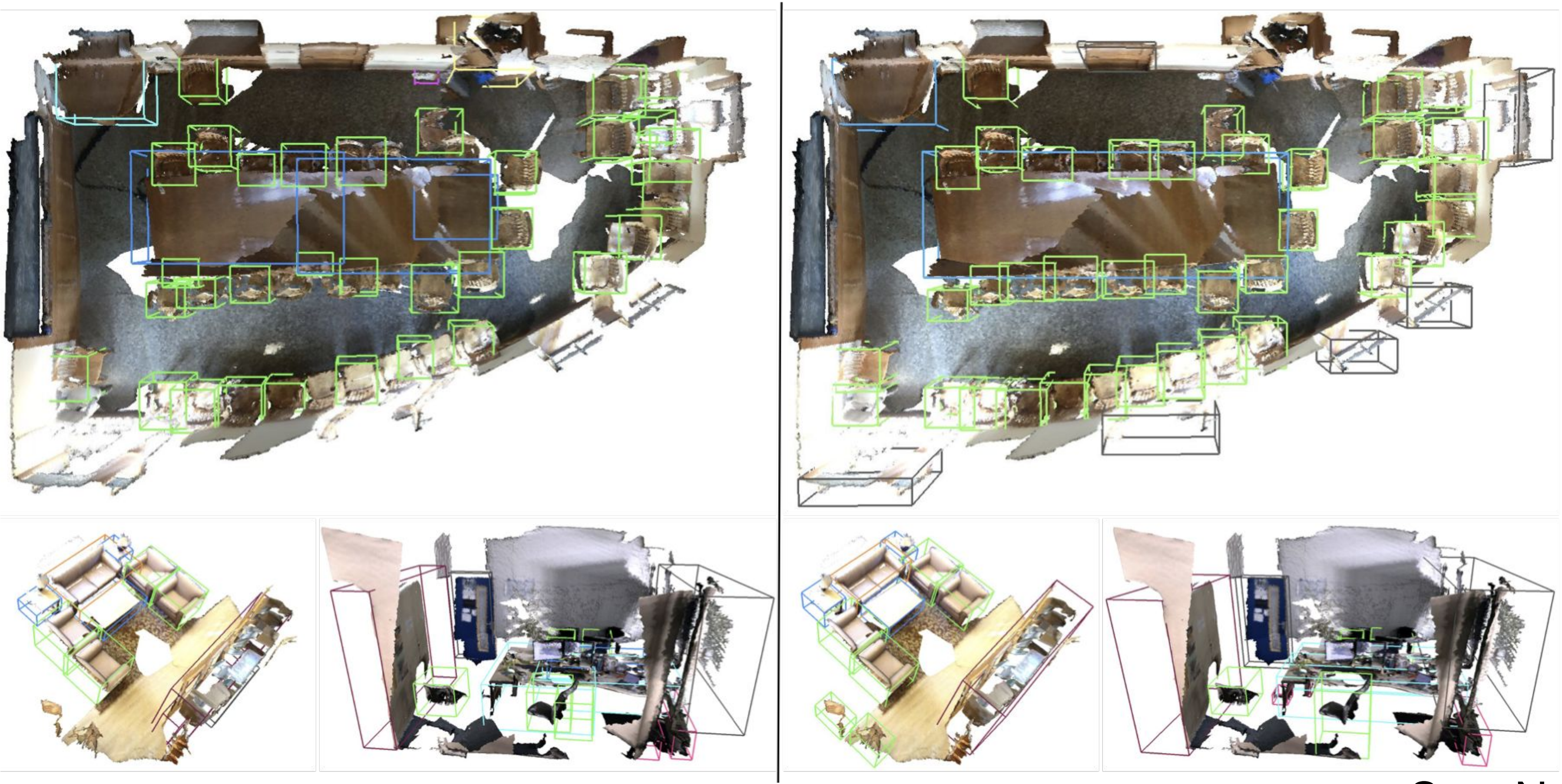
SUN RGB-D

Image of the scene | VotingNet prediction | Ground truth



SUN RGB-D

VotingNet Prediction | **Ground truth**



ScanNet

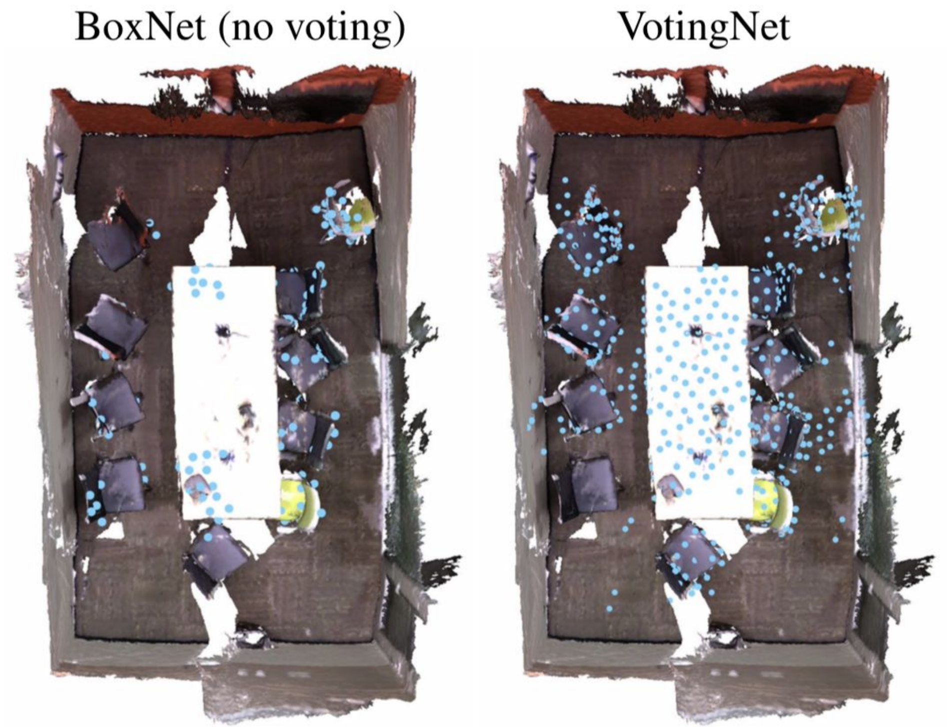
SUN RGB-D

	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**

ScanNet

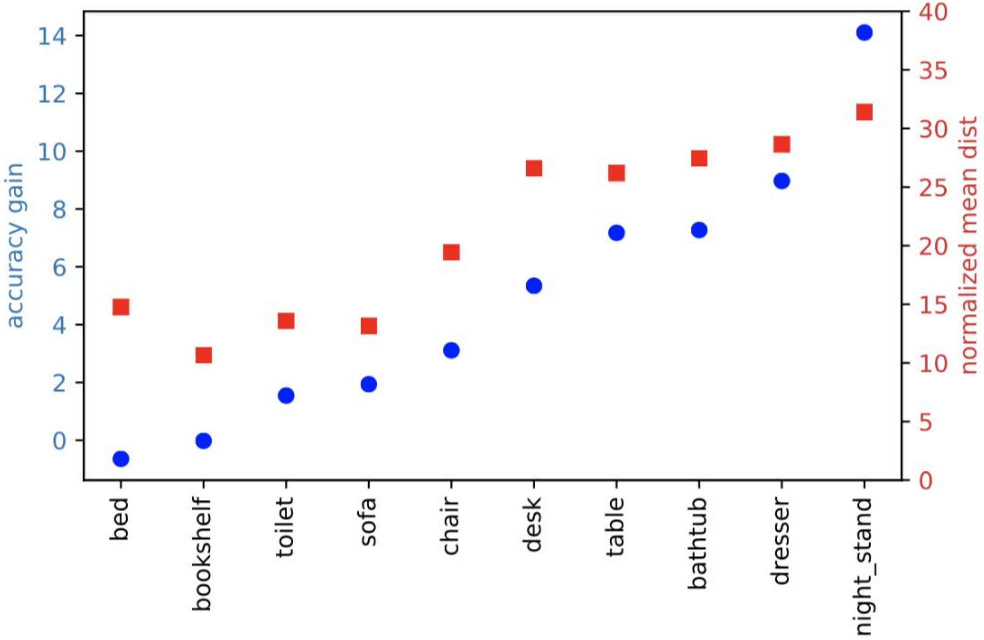
	Input	mAP@0.25	mAP@0.5
DSS [37]	Geo + RGB	15.2	6.8
MRCNN 2D-3D [10]	Geo + RGB	17.3	10.5
F-PointNet [30]	Geo + RGB	19.8	10.8
GSPN [47]	Geo + RGB	30.6	17.7
3D-SIS [11]	Geo + 1 view	35.09	18.66
3D-SIS [11]	Geo + 3 views	36.64	19.04
3D-SIS [11]	Geo + 5 views	40.22	22.53
3D-SIS [11]	Geo only	25.36	14.60
VotingNet (ours)	Geo only	**46.75**	**24.65**

To vote or not to vote?

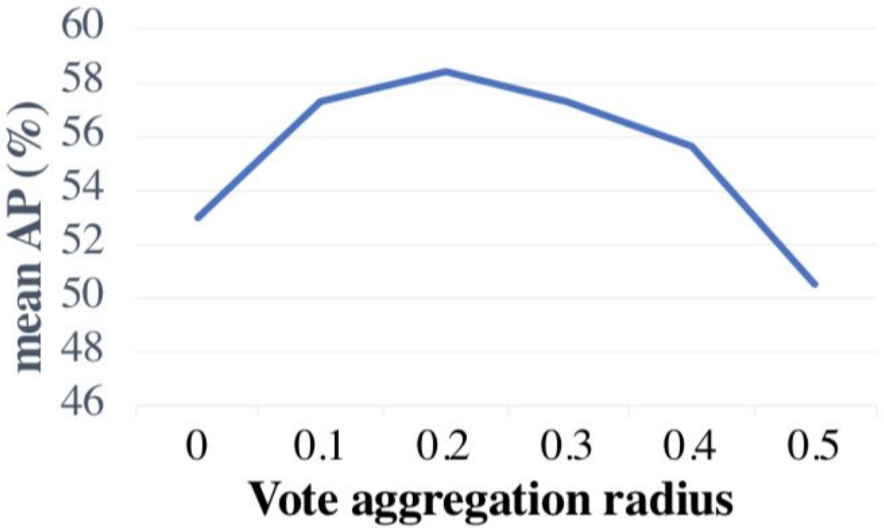


Method	3D representation	mAP@0.25 SUN RGB-D	mAP@0.25 ScanNet
DSS [37]	Volumetric	42.1	15.2
3D-SIS [11]	Volumetric	-	25.4
BoxNet (ours)	Point clouds	53.0	39.6
VotingNet (ours)	Point clouds	**57.7**	**46.8**

When does voting helps the most?

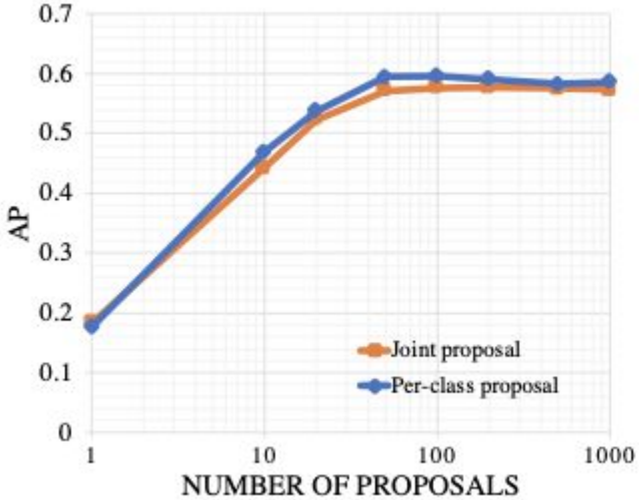


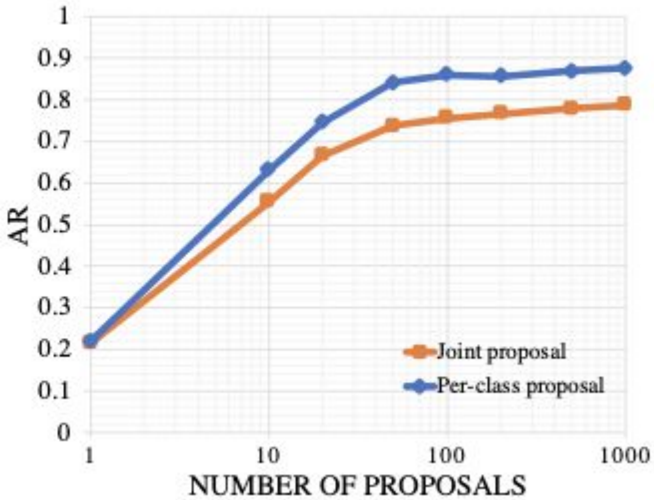
Aggregation is key



Aggregation method	mAP
Feature avg.	47.2
Feature max	47.8
Feature RBF avg.	49.0
Pointnet (avg)	56.5
Pointnet (max)	57.7

Proposal quality and runtimes

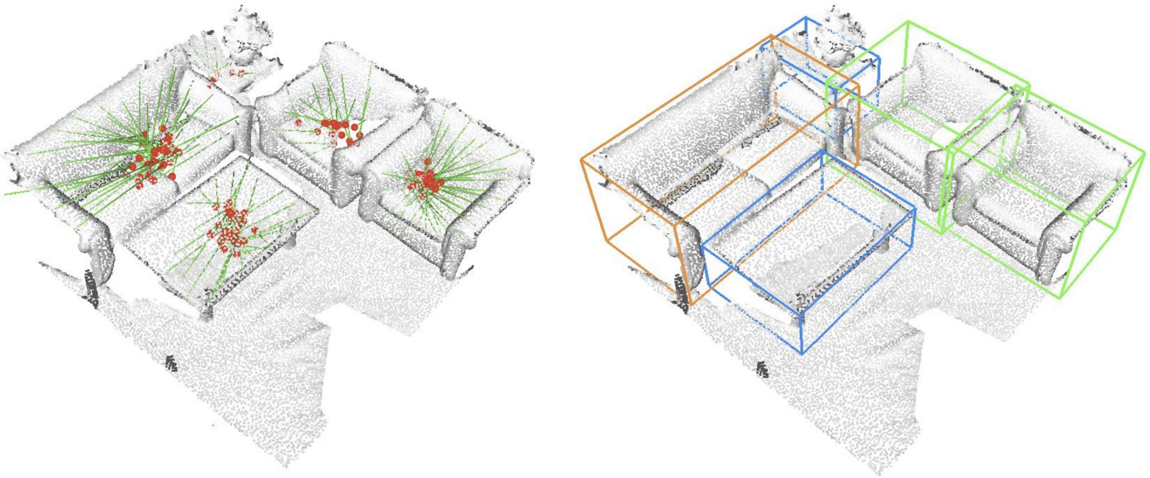




Method	Model size	SUN RGB-D	ScanNetV2
F-PointNet [30]	47.0MB	0.09s	-
3D-SIS [11]	19.7MB	-	2.85s
VotingNet (ours)	11.2MB	0.10s	0.14s

Summary

- Hough voting is back
    - Effective 3D object detection in point clouds with state-of-the-art performance on real 3D scans



Summary

- Hough voting is back
  - Effective 3D object detection in point clouds with state-of-the-art performance on real 3D scans
  - Improved context aggregation: low dimensional attention, online graph construction

- Future directions:
  - Adding color images (semantics and geometry cues)
  - Downstream tasks: extending the system to semantic / instance segmentation
  - Other use-cases suitable for voting

Thanks!

orlitany.github.io