# 3D Object Detection from Point Cloud

CS230 Deep Learning

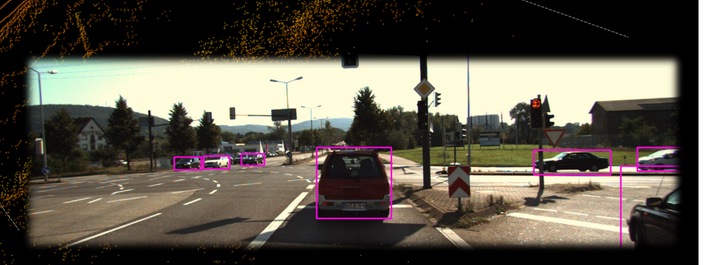
Alexander Arzhanov { aarz@stanford.edu }
*Stanford University*

youtu.be/14tuCG-tfpo

- An accurate 3D perception is indispensable for remote sensing in
  - **robotic object manipulation**
  - **augmented reality**
  - **autonomous vehicles**
- Majority of the modern **autonomous driving** systems heavily rely on **LiDAR sensors** for object tracking and collision avoidance
- However, point cloud data is

  **irregular**, **unordered**, and **sparse**

  which prevents a direct application of conv-based methods
- Approaches that first voxelize, or project point clouds into a bird's eye view are often **CPU intensive** and suffer from **information loss**
- We propose to adapt **VoteNet**[1] – an end-to-end DNN that leverages Hough voting to detect 3D objects directly from the **raw point cloud** data



## Network Architecture

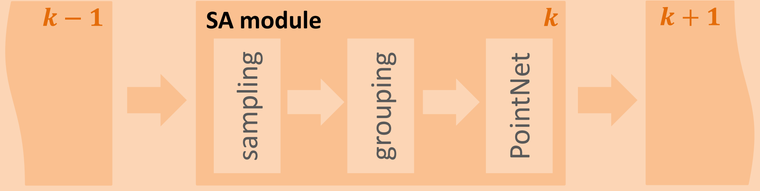
- **PointNet layer** takes as input an unordered point set

  $$P = \{\boldsymbol{p_i}\}_{i=1}^{N} \quad \text{with} \quad \boldsymbol{p_i} = [\boldsymbol{x_i}; \boldsymbol{f_i}] \in \mathbb{R}^{3+C_l}$$

  and learns a symmetric set function of the form[2]:

  $$g(\{\boldsymbol{p_i}\}) = \boldsymbol{\gamma} \circ \underset{\boldsymbol{p_i} \in P}{\mathrm{MAX}}(\{\boldsymbol{h}(\boldsymbol{p_i})\})$$

  where $h: \mathbb{R}^{C_l} \to \mathbb{R}^{D_l}$ and $\gamma: \mathbb{R}^{D_l} \to \mathbb{R}^{C_{l+1}}$ are MLP networks, MAX is channel-wise max-pooling op.: $\underbrace{\mathbb{R}^{D_l} \times \cdots \times \mathbb{R}^{D_l}}_{N} \to \mathbb{R}^{D_l}$
- **Set-abstraction (SA) module** encodes fine geometric patterns of the point cloud (PC) at different contextual scales by recursively applying *PointNet layer* on over-lapping local regions of progressively larger volume[3]:

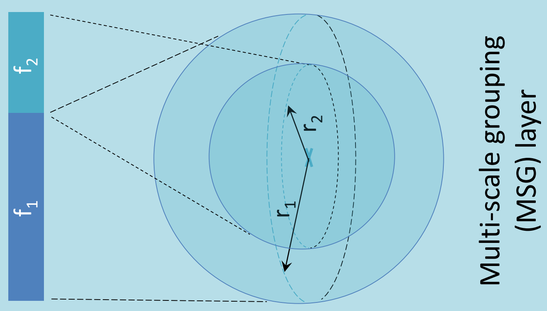



- **The feature extraction network** comprises several **SA modules** enhanced with **feature propagation (FP) layers** (skip connections) to output a total of $\boldsymbol{M}$ **seed points** enriched with $\boldsymbol{3 + C}$ deep semantic features
- **The Hough voting module** takes the $\boldsymbol{M}$ **seed points** as an input and learns a feature displacement function to output $\boldsymbol{M}$ **vote points** that cluster near object centroids
- **Object proposal and classification network** leverages an SA module to aggregate information in the ***clustered virtual points*** and generate an ***output*** $\in \mathbb{R}^{2+3+2H+4S+T}$

  with **2** objectness scores, **3** center regression values, **2H** heading bins with reg. corrections, **S** box size anchors with **3S** box size regression corrections, and **T** values for semantic classification

## Adaptation to KITTI

- KITTI 3D object detection dataset[4]:
  - **7481** annotated and **7518** test scenes of 360° LiDAR PC, RGB image, calib. matrices, *etc.*
  - **3712** scenes for training, **3769** for validation
- Preprocessing:
  - Projection of PC onto the image plane
  - Random subsampling of PC to **16,384 points**
  - Augmentations with **flips**, **rotation**, **scaling**
  - Optional extra features: **reflectance** and **height**
- Adapt network to characteristics of outdoor PC:
  - Adjust **receptive field radii** (KITTI PC spans >70m)
  - Tune **# of clusters** for feature aggregation
  - Introduce **MSG layers** for robust feature learning under non-uniform sampling density




- Final **VoteNet** parameters for to KITTI outdoor scenes

| *Modules (Input)* | *Output Dimensions* | *Grouping Clusters* | *MSG Radii (m)* | *MLP Layers* |
|---|---|---|---|---|
| $\mathbf{SA_1}$ *(PC)* | (4096, 3 + 96) | 2048 | 0.1, 0.5 | 16/16/32, 32/32/64 |
| $\mathbf{SA_2}$ *($SA_1$)* | (1024, 3 + 256) | 1024 | 0.5, 1.0 | 64/64/128, 64/96/128 |
| $\mathbf{SA_3}$ *($SA_2$)* | (512, 3 + 512) | 512 | 1.0, 2.0 | 128/196/256, 128/196/256 |
| $\mathbf{SA_4}$ *($SA_3$)* | (64, 3 + 512) | 256 | 2.0, 4.0 | 256/256/512, 256/384/512 |
| $\mathbf{FP_1}$ *($SA_3$, $SA_4$)* | (512, 3 + 512) | — | — | 512/512 |
| $\mathbf{FP_2}$ *($SA_2$, $SA_3$)* | (1024, 3 + 512) | — | — | 512/512 |

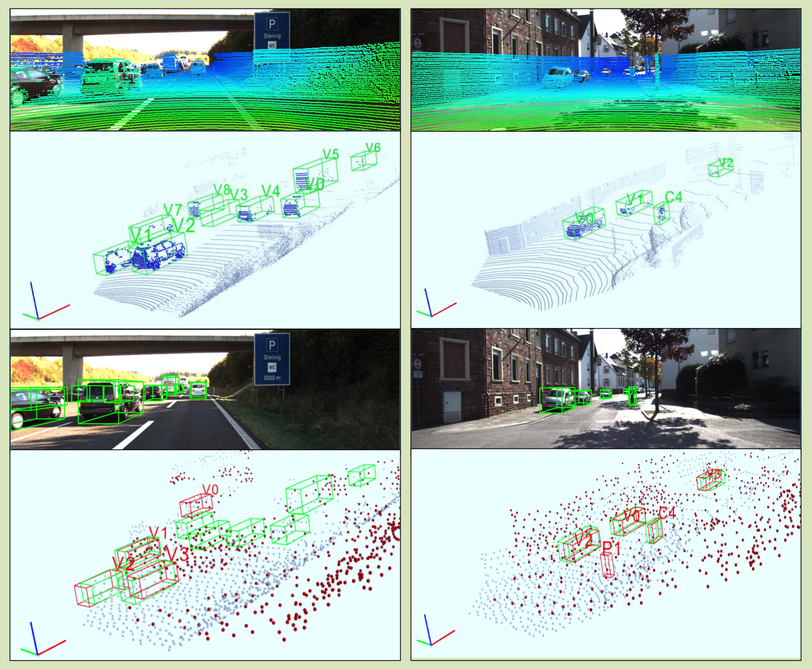
## Results

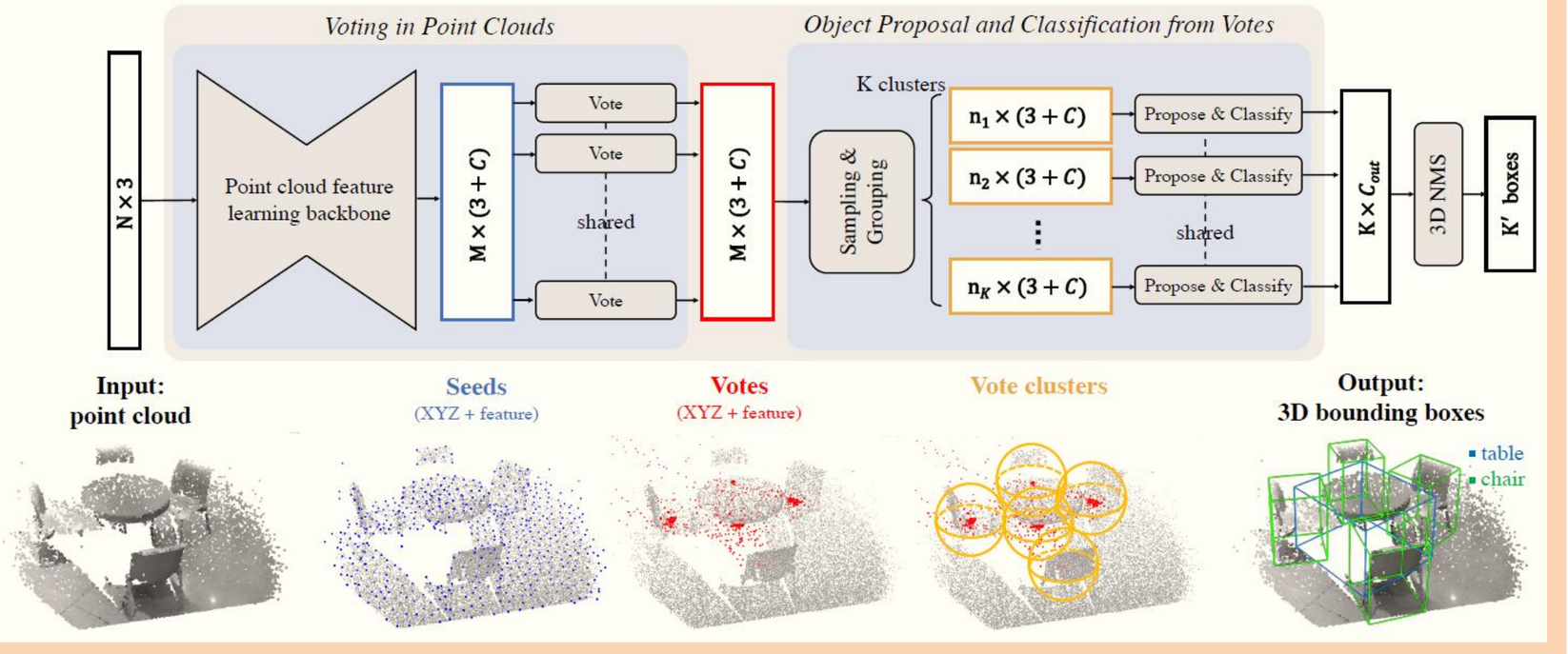
- The biggest improvement of **34.2 AP** for *Car* category, while **19.9 AP** for *Pedestrian*, and **20.5 AP** for *Cyclist*

| | *Car* | *Pedestrian* | *Cyclist* |
|---|---|---|---|
| **Original** | 21.0 | 11.3 | 0.5 |
| **Tuned** | 31.2 | 27.9 | 14.1 |
| **Tuned + MSG** | 55.2 | 31.2 | 21.0 |

Average Precision (AP) on KITTI validation split (IoU@0.25)

- Good prediction for objects close to LiDAR, even for partially occuded cases (*e.g.* parked cars)
- However, results deteriorate fast for object further away due to small number of foreground points
- The observed degree of clustering of vote points around centroids of the objects is not always found to be suffient in order to produce accurate results






courtesy of Qi C et al.[1]

## Outlook

- Compared to the recently published state-of-the-art results on KITTI 3D object detection benchmark, there are still many ways to improve
- The sparse character of the large-scale outdoor LiDAR scenes results in a poor signal-to-noise ratio, which could require a kind of point filtering, or foreground pre-segmentation step
- Alternatively, a VoteNet model could be enhanced by first predicting foreground scores for each point to weight its point features, thereby resulting in fore-ground points bearing a larger contribution to voting

References

[1] **Qi** C, **Litany** O, **He** K, and **Guibas** L, *Deep Hough voting for 3D object detection in point clouds*, ICCV, **2019**.
[2] **Qi** C, **Su** H, **Mo** K, and **Guibas** L, *PointNet: deep learning on point sets for 3D classification and segmentation*, CVPR, **2017**.
[3] **Qi** C, **Yi** L, **Su** H, and **Guibas** L, *PointNet++: deep hierarchical feature learning on point sets in a metric space*, NIPS, **2017**.
[4] **Geiger** A, **Lenz** P, and **Urtasun** R, *Are we ready for autonomous driving? the KITTI vision benchmark suite,* CVPR, **2012**.

d-fine