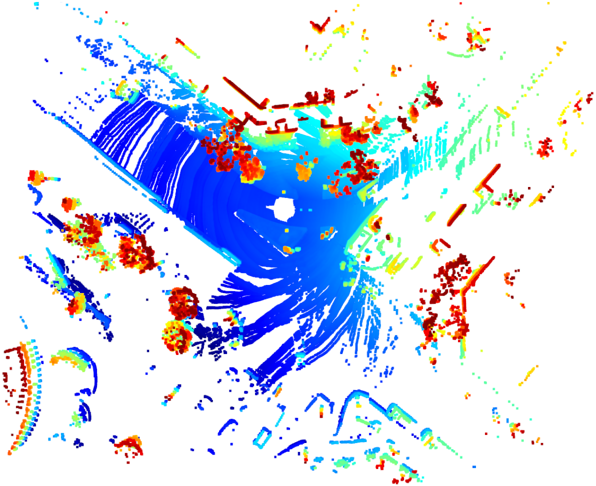
Lidar-based 3D objection detection using deep learning for autonomous vehicles applications, a review

Vincent Frémont

Workshop PPNIV 2022 – Vincent Frémont

October 23th, 2022, Kyoto, Japan

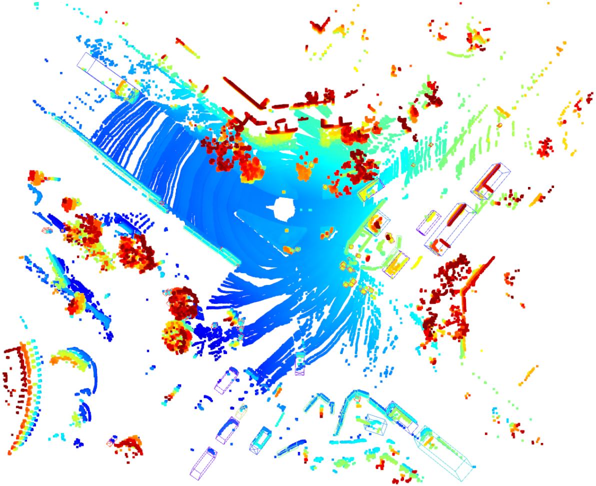
Problem Statement

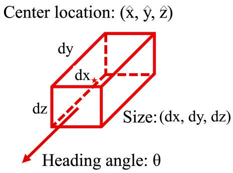


A sample point cloud of NuScenes

$$\mathcal{P} = \{\mathrm{p}_i = [x_i, y_i, z_i, f_i]\}_{i=1}^{N}$$

Problem Statement





3D Bounding Boxes

$$\mathcal{B} = \{\mathrm{b}_k = [\hat{x}_k, \hat{y}_k, \hat{z}_k, dx_k, dy_k, dz_k, \theta_k, \mathrm{cls}_k]\}_{k=1}^{M}$$

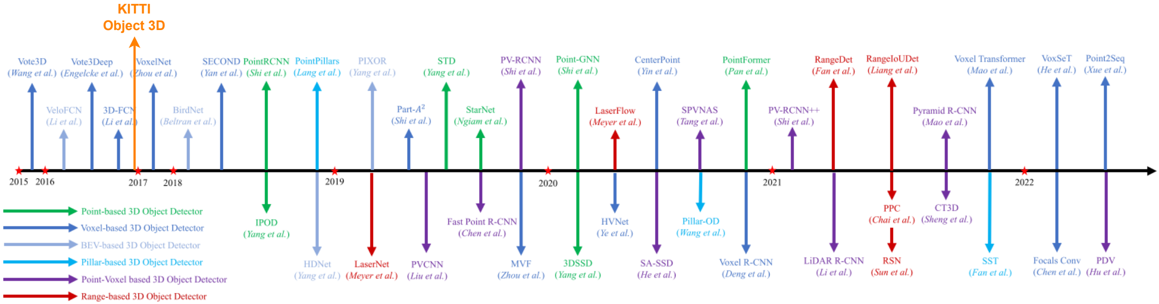
Outline

Presentation Outline

- Introduction
- Background
- Camera-LiDAR Fusion
- Practicality of 3D Object Detection Methods
- Point Cloud Densification
- Conclusion

3D Object Detection - A Brief History

… And arrived the KITTI 3D Object Detection Evaluation

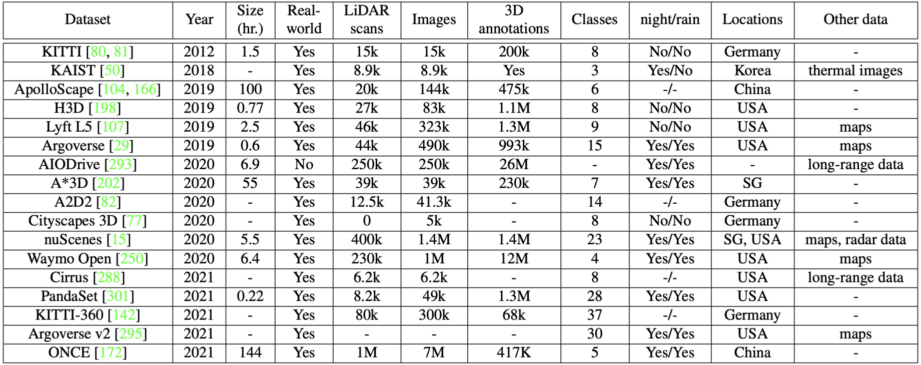


Mao et al. "3d object detection for autonomous driving: A review and new outlooks." arXiv (2022).

3D Object Detection - A Brief History

A question of datasets and benchmarks

Table 1: Datasets for 3D object detection in driving scenarios.

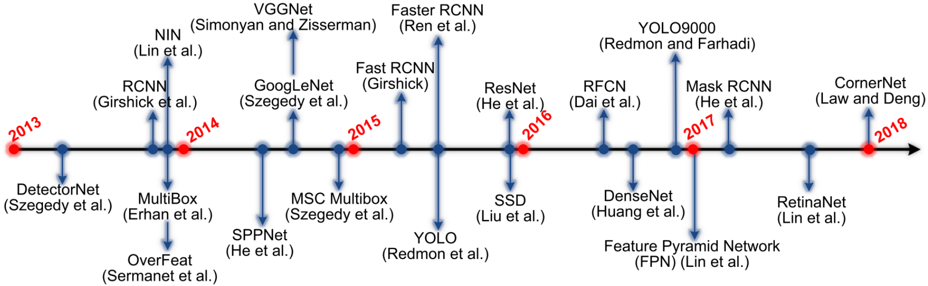


Dataset	Year	Size (hr.)	Real-world	LiDAR scans	Images	3D annotations	Classes	night/rain	Locations	Other data
KITTI [80, 81]	2012	1.5	Yes	15k	15k	200k	8	No/No	Germany	-
KAIST [50]	2018	-	Yes	8.9k	8.9k	Yes	3	Yes/No	Korea	thermal images
ApolloScape [104, 166]	2019	100	Yes	20k	144k	475k	6	-/-	China	-
H3D [198]	2019	0.77	Yes	27k	83k	1.1M	8	No/No	USA	-
Lyft L5 [107]	2019	2.5	Yes	46k	323k	1.3M	9	No/No	USA	maps
Argoverse [29]	2019	0.6	Yes	44k	490k	993k	15	Yes/Yes	USA	maps
AIODrive [293]	2020	6.9	No	250k	250k	26M	-	Yes/Yes	-	long-range data
A*3D [202]	2020	55	Yes	39k	39k	230k	7	Yes/Yes	SG	-
A2D2 [82]	2020	-	Yes	12.5k	41.3k	-	14	-/-	Germany	-
Cityscapes 3D [77]	2020	-	Yes	0	5k	-	8	No/No	Germany	-
nuScenes [15]	2020	5.5	Yes	400k	1.4M	1.4M	23	Yes/Yes	SG, USA	maps, radar data
Waymo Open [250]	2020	6.4	Yes	230k	1M	12M	4	Yes/Yes	USA	maps
Cirrus [288]	2021	-	Yes	6.2k	6.2k	-	8	-/-	USA	long-range data
PandaSet [301]	2021	0.22	Yes	8.2k	49k	1.3M	28	Yes/Yes	USA	-
KITTI-360 [142]	2021	-	Yes	80k	300k	68k	37	-/-	Germany	-
Argoverse v2 [295]	2021	-	Yes	-	-	-	30	Yes/Yes	USA	maps
ONCE [172]	2021	144	Yes	1M	7M	417K	5	Yes/Yes	China	-

Mao et al. "3d object detection for autonomous driving: A review and new outlooks." arXiv (2022).

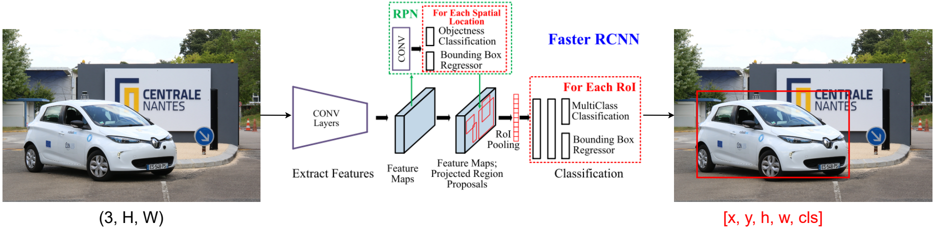
3D Object Detection - A Brief History

Inspiration from 2D Objects Detection



Liu et al. "Deep learning for generic object detection: A survey." IJCV (2020)

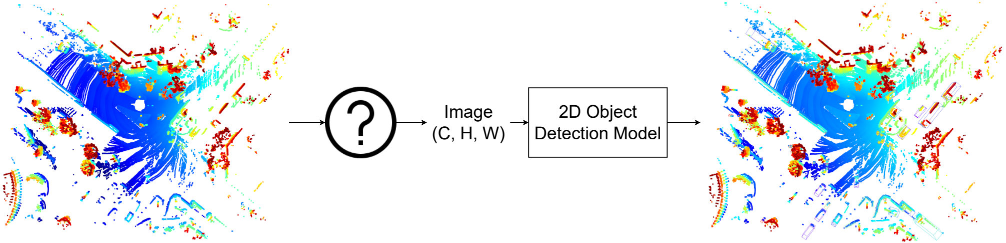
2D Object Detection - In a Nutshell



Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." NeurIPS (2015).

- **Feature Extraction** using a 2D CNN
- **Proposals Generation** At each location of Feature Maps, an Region Proposal Network (a small CNN)
  - Predict $2k$ classification scores representing positive/ negative probability of k anchors at this location
  - Predict $4k$ floating values representing difference between k anchors at this location and their associated ground truth
- **Proposals Refinement** For each *positive* anchor,
  - ROI Pooling to obtain ROI feature vector
  - 2 sibling FFN map ROI feature vector to its class probability and its difference w.r.t its ground truth

Moving to 3D



$\mathcal{P} = \{\mathbf{p}_i = [x_i, y_i, z_i, f_i]\}_{i=1}^{N}$

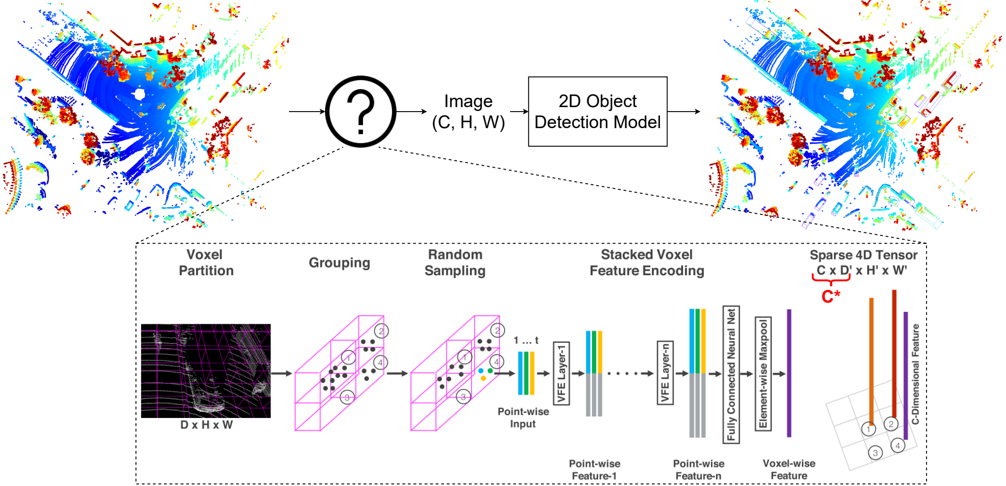
$\mathcal{B} = \{\mathbf{b}_k = [\hat{x}_k, \hat{y}_k, \hat{z}_k, dx_k, dy_k, dz_k, \theta_k, \mathrm{cls}_k]\}_{k=1}^{M}$

Central question of *early* 3D Object Detection: computing image-like representation of point clouds

Voxel-based Methods - VoxelNet

$$\mathcal{P} = \{\mathbf{p}_i = [x_i, y_i, z_i, f_i]\}_{i=1}^{N}$$

$$\mathcal{B} = \{\mathbf{b}_k = [\hat{x}_k, \hat{y}_k, \hat{z}_k, dx_k, dy_k, dz_k, \theta_k, \text{cls}_k]\}_{k=1}^{M}$$



?

Image (C, H, W)

2D Object Detection Model

Voxel Partition

Grouping

Random Sampling

Stacked Voxel Feature Encoding

Sparse 4D Tensor

$\underbrace{C \times D'}_{C^*} \times H' \times W'$

D x H x W

1 ... t

Point-wise Input

VFE Layer-1

Point-wise Feature-1

VFE Layer-n

Point-wise Feature-n

Fully Connected Neural Net

Element-wise Maxpool

Voxel-wise Feature

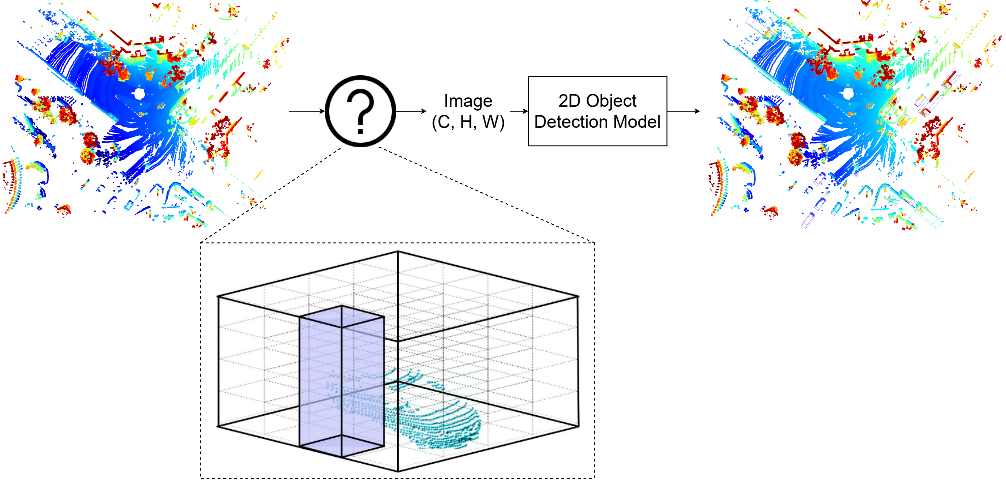
C-Dimensional Feature

Zhou et al. "Voxelnet: End-to-end learning for point cloud based 3d object detection." CVPR (2018).

Voxel-based Methods - PIXOR

$\mathcal{P} = \{\mathbf{p}_i = [x_i, y_i, z_i, f_i]\}_{i=1}^{N}$

$\mathcal{B} = \{\mathbf{b}_k = [\hat{x}_k, \hat{y}_k, \hat{z}_k, dx_k, dy_k, dz_k, \theta_k, \text{cls}_k]\}_{k=1}^{M}$

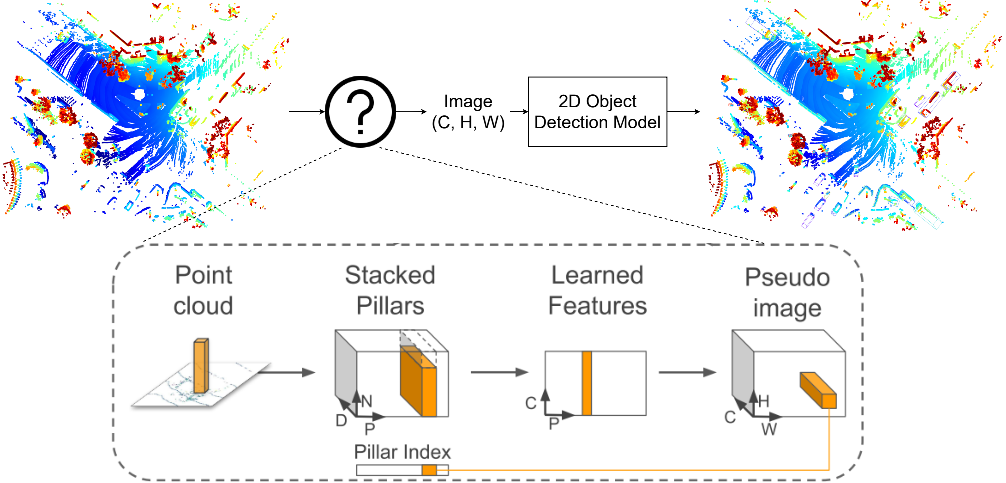


Yang et al. "Pixor: Real-time 3d object detection from point clouds." CVPR (2018).

Voxel-based Methods - PointPillars

$$\mathcal{P} = \{\mathbf{p}_i = [x_i, y_i, z_i, f_i]\}_{i=1}^{N}$$

$$\mathcal{B} = \{\mathbf{b}_k = [\hat{x}_k, \hat{y}_k, \hat{z}_k, dx_k, dy_k, dz_k, \theta_k, \text{cls}_k]\}_{k=1}^{M}$$

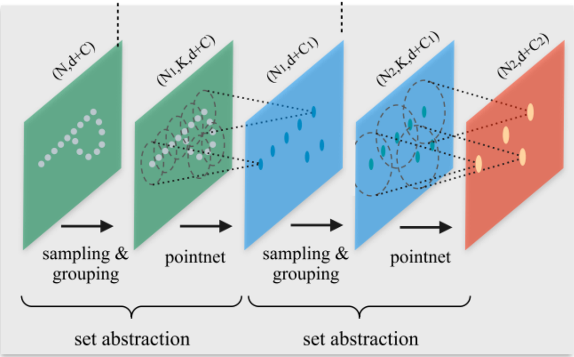


Lang et al. "Pointpillars: Fast encoders for object detection from point clouds." CVPR (2019).

Voxel-based Methods - Advantages & Drawbacks

- Advantages
  - Grid-based representation compatible with the Convolution operators
- Drawbacks
  - Information loss due to voxelization
    - To be solved by Point-based methods
  - Waste computations on empty locations of Bird-Eye View images of point clouds

Point-based Methods - PointNet



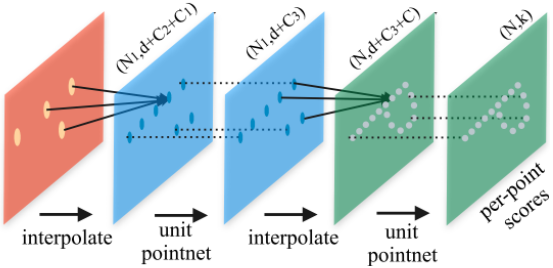
Qi et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." NeurIPS (2017).

Let $x_c \in \mathbb{R}^3$ denote the location of a point of interest which has N points $x_i(i = 1, ..., N)$ in its neighborhood. $f_{(\cdot)}$ denotes a point feature

$$f_c = \text{FFN}\left(\max_i \text{FFN}\left([f_i, x_i - x_c]\right)\right)$$

Set Abstraction ∼ Convolution on Set

Point-based Methods - PointNet



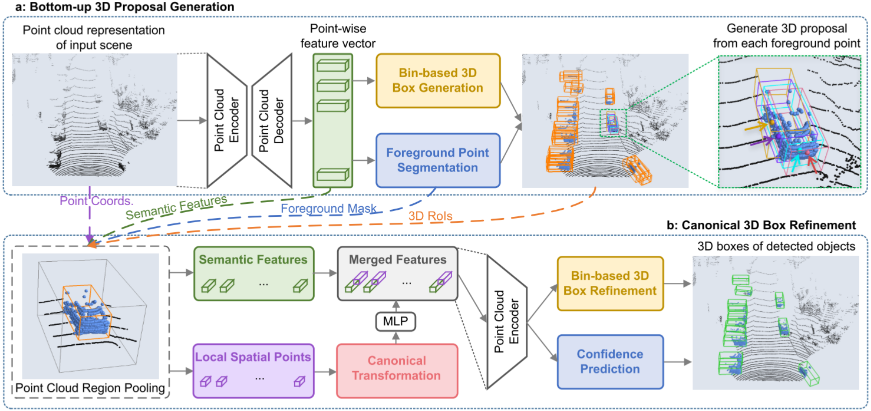
Qi et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." NeurIPS (2017).

Let $\mathrm{x}_c \in \mathbb{R}^3$ denote the location of a point of interest which has N points $\mathrm{x}_i (i = 1, ..., N)$ in its neighborhood. $\mathrm{f}_{(\cdot)}$ denotes a point feature

$$\mathrm{f}_c = \frac{\sum_i w_i \mathrm{f}_i}{\sum_i w_i} \text{ where } w_i = \frac{1}{d(\mathrm{x}_c, \mathrm{x}_i)^p}$$

Feature Propagation Layer $\sim$ Convolution Transpose

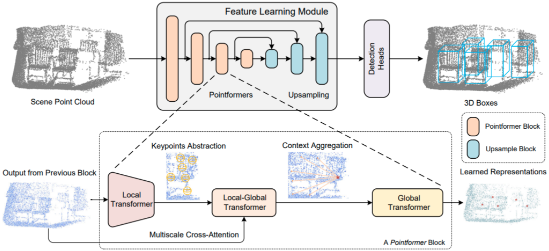
Point-based Methods - PointRCNN


a: Bottom-up 3D Proposal Generation
Point cloud representation of input scene
Point Cloud Encoder
Point Cloud Decoder
Point-wise feature vector
Bin-based 3D Box Generation
Foreground Point Segmentation
Generate 3D proposal from each foreground point
Point Coords.
Semantic Features
Foreground Mask
3D RoIs
b: Canonical 3D Box Refinement
Point Cloud Region Pooling
Semantic Features
Merged Features
MLP
Local Spatial Points
Canonical Transformation
Point Cloud Encoder
Bin-based 3D Box Refinement
Confidence Prediction
3D boxes of detected objects

Shi et al "Pointrcnn: 3d object proposal generation and detection from point cloud." CVPR (2019).

- Point Cloud Encoder $\equiv$ a stack of Set Abstraction Layers
- Point Cloud Decoder $\equiv$ a stack of Feature Propagation Layers

Point-based Methods - PointFormer



Pan et al. "3d object detection with pointformer." CVPR (2021).

Let $x_c \in \mathbb{R}^3$ denote the location of a point of interest which has N points $x_i (i = 1, ..., N)$ in its neighborhood.
$f_{(\cdot)}$ denotes a point feature

- Set Abstraction Layer:

$$f_c = \text{FFN}\left(\max_i \text{FFN}\left([f_i, x_i - x_c]\right)\right)$$

- Local Transformer

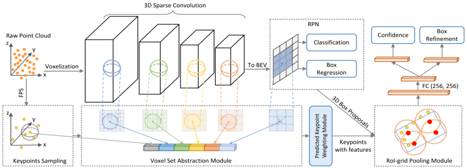
$$f_c^l = \text{CrossAttention}\left(f_c^{l-1}; [f_i, x_i - x_c]_{i=1}^N\right)$$

  - Key: features of neighbor points $\{f_i\}_{i=1}^N$
  - Query: features of the point of interest f_c^l calculated by the previous layer $l - 1$

Point-based Methods - Advantages & Drawbacks

- Advantages
  - Avoiding information loss due to voxelization
  - Operating directly on points → suitable for in-door scenes where objects density is high
- Drawbacks
  - Information loss due to point cloud subsampling
  - Point cloud query ops (furthest points sampling, nearest neighbors query) induces large computational overhead
    - Challenging to meet the real-time requirement on large scale out-door point clouds

Combining Voxels and Points - PV-RCNN

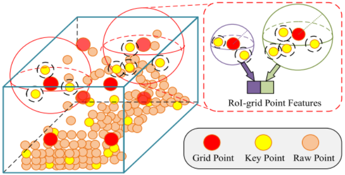


Shi et al. "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection." CVPR (2020).

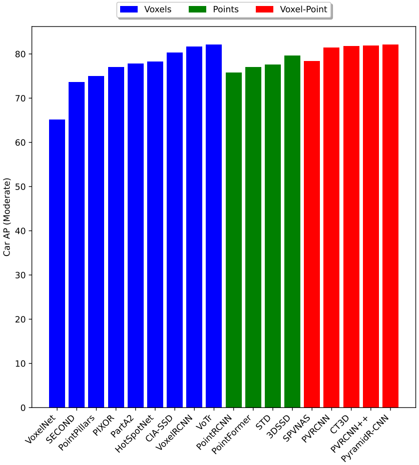
- **Feature Extraction** & **Proposals Generation** by Voxel-based method (VoxelNet)
- **Proposals Refinement** by Point-based method combined with a 3D version of ROIAlign (**He et al. "Mask r-cnn." ICCV (2017)**)

$$\text{Key points feature} = \text{SetAbstraction}\left(\text{raw points feature}\right)$$

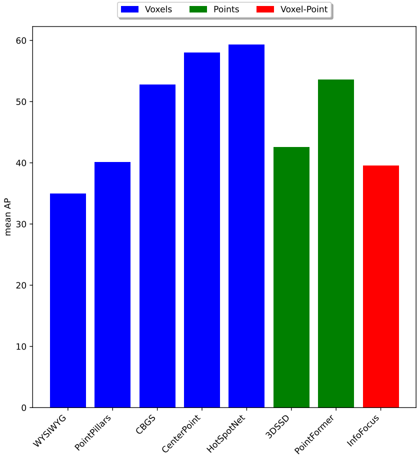
$$\text{Grid points feature} = \text{SetAbstraction}\left(\text{Key points feature}\right)$$



Performance Comparison

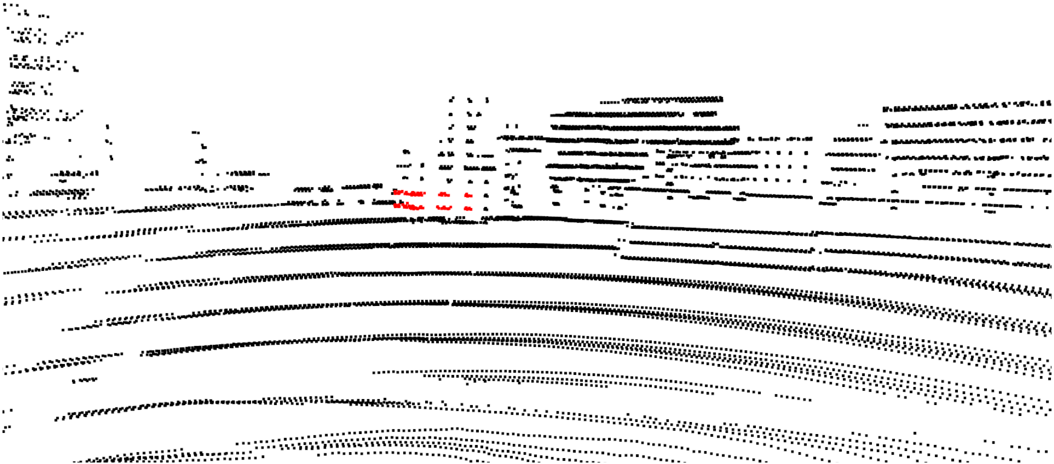


(a) KITTI



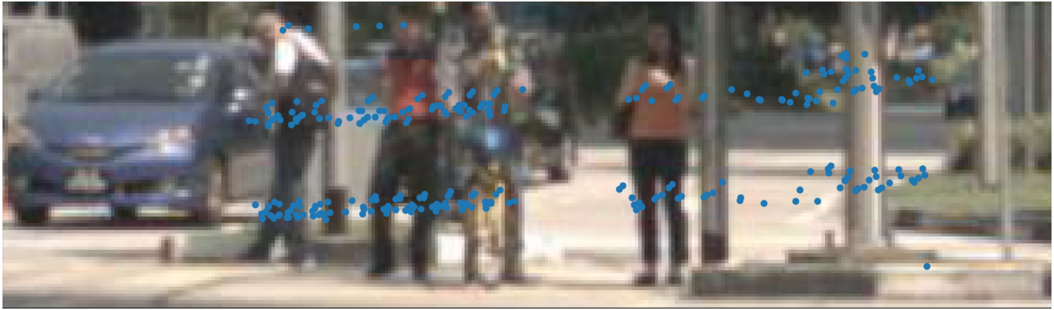
(b) NuScenes

LiDARs Can't See Texture



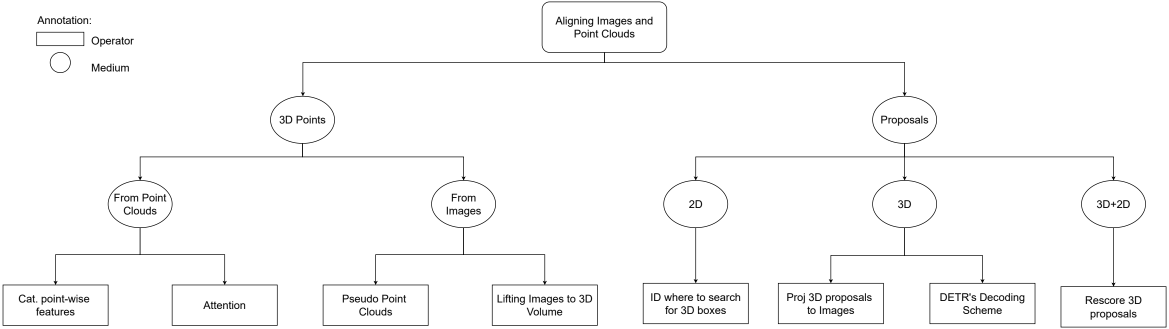
A collection of groups of points that have similar appearance
How many pedestrians?

Cameras to The Rescue

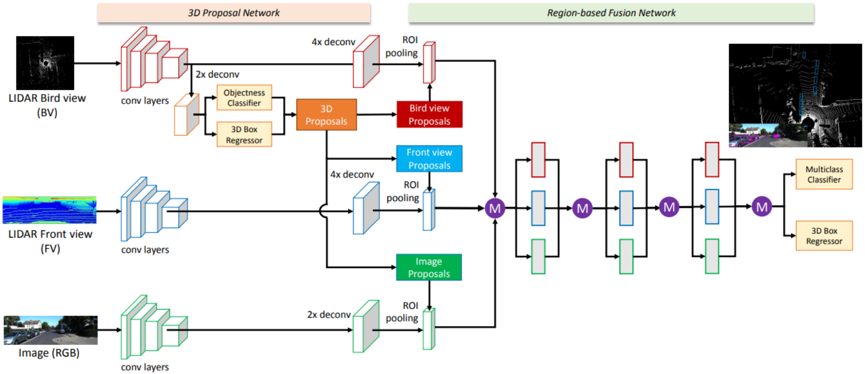


Projection of the collection of groups of points onto vehicle's camera

Taxonomy of Camera-LiDAR Fusion



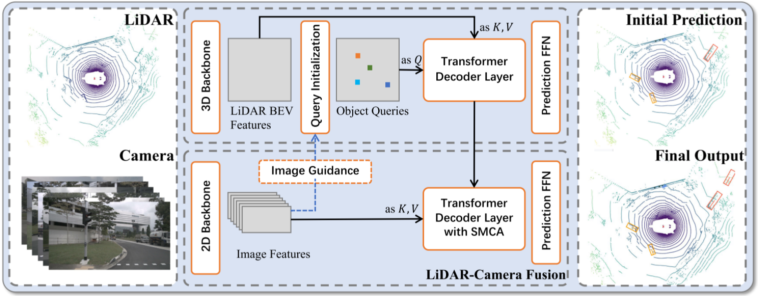
3D Proposals - Projecting to Images



Chen et al. "Multi-view 3d object detection network for autonomous driving." CVPR (2017).

- 3D Proposals are generated in Bird-Eye View (BEV), then projected to: BEV, (LiDAR) Front View, Image
- Proposal's view-dependent feature is obtained by ROI Pooling based on its projection

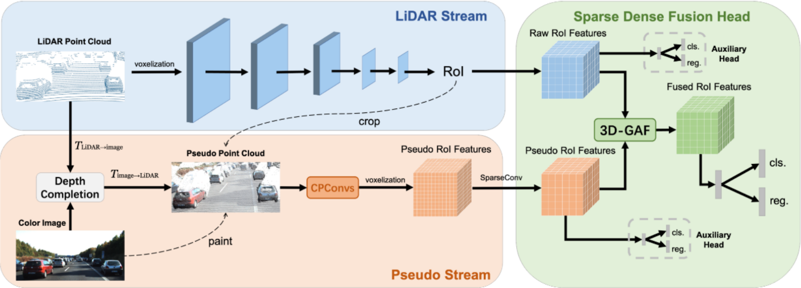
3D Proposals - DETR's Decoding Scheme



Bai et al. "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers." CVPR (2022).

- In LiDAR branch, objects are detected using DETR (Carion et al. "End-to-end object detection with transformers." ECCV, (2020).)
  - Object Query $= \left\{ q_i^{\textbf{init}} \in \mathbb{R}^d \right\}_{i=1}^{N}$. $q_i^{\textbf{init}}$ is a learnable vector
  - $q_i^{\textbf{lidar}} = \text{CrossAttn}\left(q_i^{\textbf{init}}, I^{BEV}\right)$. I^{BEV} is the pseudo-BEV image computed by PointPillars.
- In Image branch,
  - 2D CNN extracts image features I^{img}
  - $q_i^{\textbf{fuse}} = \text{CrossAttn}\left(q_i^{\textbf{lidar}}, I^{img}\right)$
- $q_i^{\textbf{fuse}}$ is decoded into a 3D bounding box by a-shared FFN

3D Points From Images - Pseudo Point Clouds



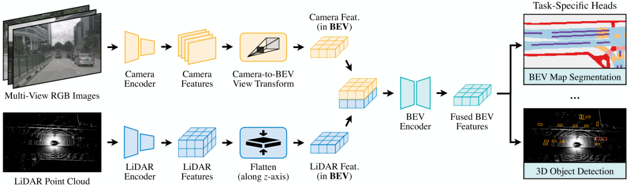
Wu et al. "Sparse Fuse Dense: Towards High Quality 3D Detection with Depth Completion." CVPR (2022).

- **Feature Extraction** & **Proposals Generation** by Voxel-based backbone (SECOND)
- **Proposals Refinement** by
  - Pooling raw points → voxelize → extract features for voxel grid using Convolution
  - Pooling pseudo points → voxelize → extract features for voxel grid using Convolution
  - Features $\mathsf{F}_i^{\square}$ at location i in two voxel grid above are fused by a weighting scheme

$$\left(w_i^{\text{raw}}, w_i^{\text{pse}}\right) = \sigma\left(\text{MLP}\left(\text{CAT}\left(\mathsf{F}_i^{\text{raw}}, \mathsf{F}_i^{\text{pse}}\right)\right)\right)$$

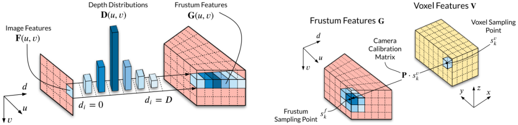
$$\mathsf{F}_i = \text{MLP}\left(\text{CAT}\left(w_i^{\text{raw}}\mathsf{F}_i^{\text{raw}}, w_i^{\text{pse}}\mathsf{F}_i^{\text{pse}}\right)\right)$$

3D Points From Images - Lifting Images to 3D



Liu et al. "BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation." arXiv (2022).

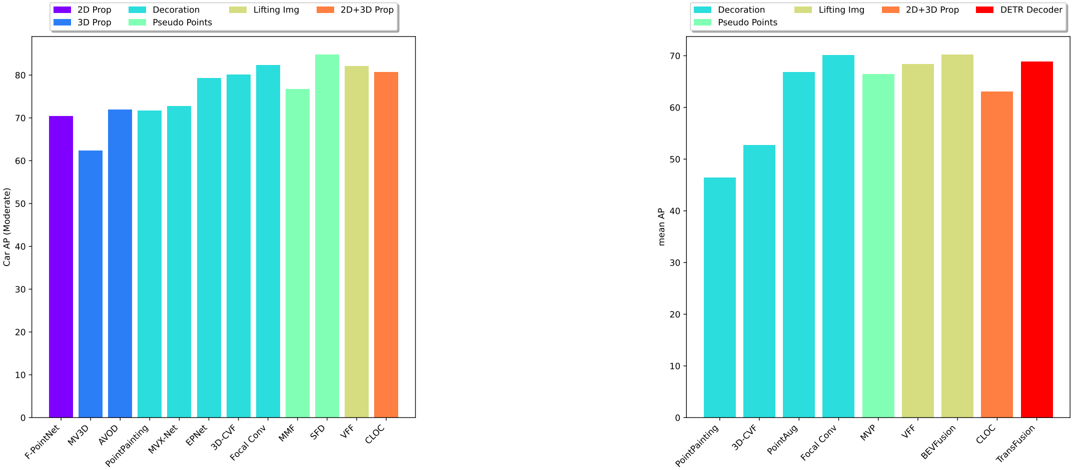
Lifting images to 3D volume by predicting a categorical depth distribution for every pixel



Reading et al. "Categorical depth distribution network for monocular 3d object detection." CVPR (2021).

- Image Features F: $W_F \times H_F \times C$
- Frustum Features G: $W_F \times H_F \times D \times C$
- Voxel Features V: $X \times Y \times Z \times C$
  - (X, Y) are linked to (W_F, H_F) by camera projection matrix
  - Z is linked to D by discretization method (e.g., $d = z/\delta D$)

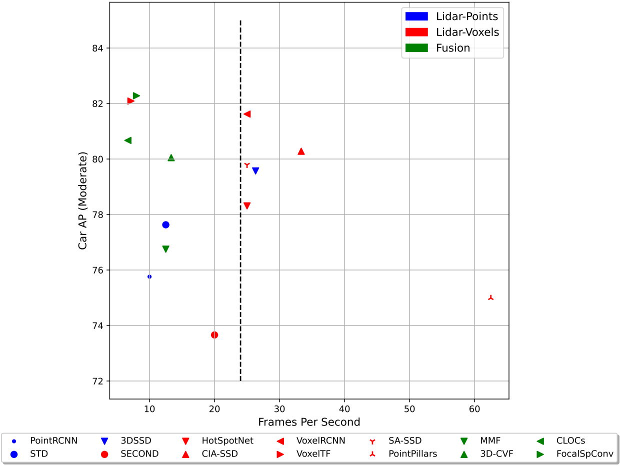
Performance Comparison



(a) KITTI

(b) NuScenes

Accuracy-Speed Tradeoff



Performance of 3D Detectors on KITTI (test) with respect to their inference speed

- Dash vertical line shows the real-time threshold (24 Hz)
- Performances are reported by their respective papers → not tested on the same hardware
- Non-exhausting list (most fusion methods don't report their inference time)

Observations:

- Camera-LiDAR Fusion yields the strongest performance at the cost of low inference speed
- Voxel-based methods spread over the entire accuracy/speed spectrum
  - High accuracy: Voxel Transformer
  - High frame-rate: PointPillars
  - Best balance: Voxel RCNN

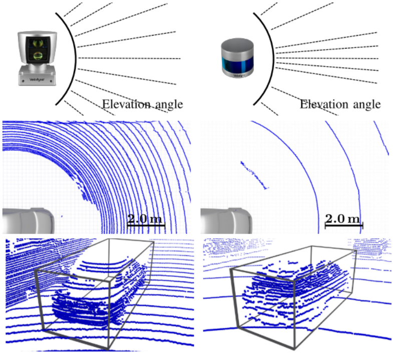
Impact of Changes in LiDAR Resolution

Performance of PointRCNN across multiple datasets. The evaluation is performed for **class Car** only and is measured by AP_{3D}. Best and worst generalization (domain adaptation) in each setting are marked by red and blue, respectively. In domain performance is indicated by **bold** font.

Setting	Source/Target	KITTI	Argoverse	NuScenes	Lyft	Waymo
0-30m	KITTI	**84.9**	34.7	14.9	54.2	14.0
	Argoverse	46.8	**63.3**	26.9	69.5	44.4
	NuScenes	13.9	26.0	**42.8**	43.8	43.4
	Lyft	45.2	54.0	25.4	**88.5**	70.9
	Waymo	15.0	48.1	24.0	76.2	**87.2**
30-50m	KITTI	**51.4**	19.0	4.5	34.5	21.4
	Argoverse	11.8	**39.5**	9.1	39.1	42.1
	NuScenes	3.8	6.4	**4.1**	18.9	29.2
	Lyft	16.6	21.8	9.1	**62.7**	55.5
	Waymo	9.3	18.8	9.1	51.4	**68.8**
30-50m	KITTI	**12.0**	3.0	0.0	9.6	12.0
	Argoverse	1.3	**6.9**	0.0	14.5	23.0
	NuScenes	1.5	2.3	**9.1**	5.3	15.2
	Lyft	4.6	3.9	0.0	**33.1**	27.5
	Waymo	1.8	5.6	0.0	21.3	**41.1**

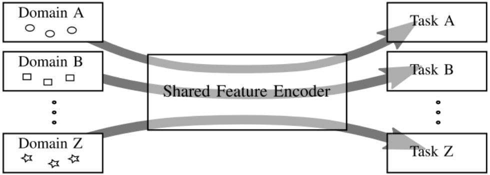
Wang et al. "Train in germany, test in the usa: Making 3d object detectors generalize." CVPR (2020).

Robustness to LiDAR Resolution by A Mixture of Datasets



(a) Velodyne-HDL64 (b) Velodyne-VLP32

(a) 64-channel point cloud comapred to 32-channel



(b) Alternating Training Method

Rist et al "Cross-sensor deep domain adaptation for LiDAR detection and segmentation." IV (2019).

Robustness to LiDAR Resolution by A Mixture of Datasets

Table: Datasets specification

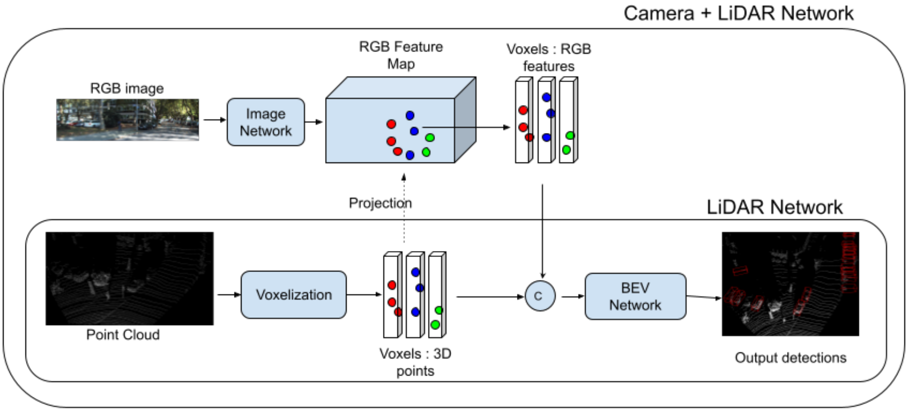
Dataset [# channels]	Task	# Annotated Frames		
		Train	Val.	Test
KITTI Object (**K**) [64]	Obj. Det.	3712	3769	7518
LiDAR Semantic (**S**) [32]	Sem. Seg.	340 000	12 261	22 983
LiDAR Multitask (**M**) [32]	Obj. Det. + Sem. Seg.	1047	226	441

Table: Performance of different pre-training dataset

Pre-train on	**Multi-task (M)**[32]			
	Detection AP			SemSeg
	Hard	Mod	Easy	mIoU
No pre-training	60.3	67.8	69.5	67.0
KITTI Object (**K**) [64]	72.4	81.4	83.3	46.4
LiDAR Semantic (**S**) [32]	41.6	47.6	50.1	69.1
(**K**) [64] + (**S**) [32]	**74.8**	**82.0**	**84.8**	**69.5**

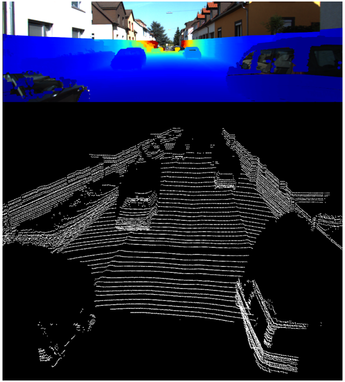
LiDAR Resolution-Agnostic Object Detection

- Architecture based on PointPillars **[Lang et al. CVPR 2019]**
- Output inspired from CenterNet **[Zhou et al. CoRR 2019]** and anchor-free

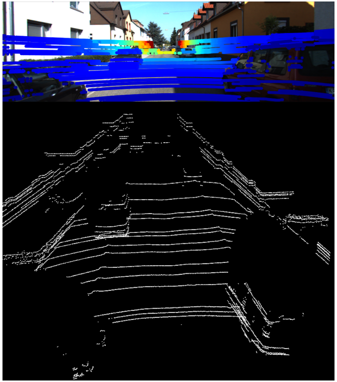


Theodose et al. "A Deep Learning Approach for LiDAR Resolution-Agnostic Object Detection." T-ITS (2021).

First Modification: Layers sub-sampling



(a) Original Point Cloud (64 layers)



(b) Reduced Point Cloud (28 layers)

Second Modification: Objects Representation

Classical box representation (x, y, w, l, θ) to Gaussian Representation $\mathcal{N}(\mu, \Sigma)$ as

$$\mu = [x, y]^T \in \mathbb{R}^2, \Sigma = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

with

$$a = \frac{\cos^2(\theta)}{2\sigma_w^2} + \frac{\sin^2(\theta)}{2\sigma_l^2},\ b = -\frac{\sin(2\theta)}{4\sigma_w^2} + \frac{\sin(2\theta)}{4\sigma_l^2},\ c = \frac{\sin^2(\theta)}{2\sigma_w^2} + \frac{\cos^2(\theta)}{2\sigma_l^2},\ \sigma_w = \frac{w}{3},\ \sigma_l = \frac{l}{3},$$

Loss Function Regression part:

- (x, y): Smooth L1

$$f(x_{pred}, x_{gt}) = \begin{cases} 0.5(x_{pred} - x_{gt})^2 & \text{if } |x_{pred} - x_{gt}| < 1 \\ |x_{pred} - x_{gt}| - 0.5 & \text{otherwise} \end{cases}$$

- (w, l, θ): simplified Bhattacharyya distance

$$D_B(\Sigma_{pred}, \Sigma_{gt}) = \ln \frac{\det \Sigma}{\sqrt{\det \Sigma_{pred} \det \Sigma_{gt}}}, \Sigma = \frac{\Sigma_{pred} + \Sigma_{gt}}{2}$$

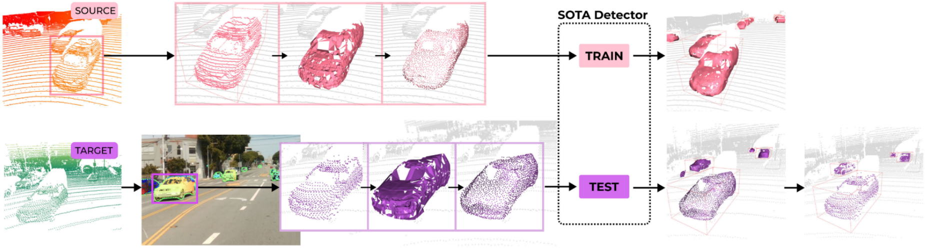
Results

Table: Evaluation for BEV detection on the KITTI validation set, nuScenes Mini and Pandaset datasets. Results in Average Precision (%). The last number in dataset identifier indicates the IoU threshold used for computing the scores.

Exp. ID	KITTI 64 layers 0.7			KITTI 8 layers 0.7			nuScenes	Pandaset	Mean
	Easy	Moderate	Hard	Easy	Moderate	Hard	0.5	0.5	Datasets
#1 CL-64-Std	75.02	59.22	54.71	41.81	29.48	27.29	9.09	11.26	29.04
#2 CL-64-G	72.07	53.10	53.31	42.11	29.59	27.47	9.09	15.31	29.23
#3 CL-Var-Std	77.46	58.49	54.50	61.57	40.46	38.83	15.65	14.39	35.11
#4 CL-Var-G	70.18	51.90	52.55	57.14	38.61	36.34	16.61	7.41	31.56
#5 CL-8-Std	71.55	58.72	54.57	65.30	46.96	41.10	16.26	4.73	33.43
#6 CL-8-G	65.55	48.19	47.92	64.34	45.12	40.06	16.35	9.09	32.29
#7 L-64-Std	85.35	75.16	71.13	27.55	20.95	18.21	37.69	49.35	46.62
#8 L-64-G	84.93	75.02	71.59	27.21	18.99	16.62	50.12	52.63	50.21
#9 L-Var-Std	86.46	75.69	74.43	58.61	40.34	35.81	47.86	49.00	55.16
#10 L-Var-G	85.90	75.45	72.30	58.16	40.56	35.48	66.91	52.26	60.44
#11 L-8-Std	48.54	44.18	44.53	67.65	48.26	43.62	55.04	16.82	42.69
#12 L-8-G	43.85	42.18	41.95	66.32	47.18	42.83	44.43	11.58	37.69
PointPillars	89.65	87.17	84.37	46.99	32.34	27.85	29.38	30.47	45.66
PV-RCNN	90.26	88.04	87.39	40.02	28.66	26.48	32.96	39.73	48.24

The CL-experiments run at 35 ms and the L-experiments run at 20 ms on a computer equipped with a GPU NVidia 1080Ti and a CPU Intel Core i7−7700K.

Robustness to LiDAR Resolution by Standardizing the Number of Points



Tsai et al. "See Eye to Eye: A Lidar-Agnostic 3D Detection Framework for Unsupervised Multi-Target Domain Adaptation." RA-L (2022).

- Densify point clouds of target domain so that they have the same number of points as source domain's
  - Isolating object points using projection of 3D points to image and instance masks (e.g., by MaskRCNN)
  - Surface Completion using the resulting object points and Ball-Pivoting algorithm
  - Sampling more points from the resulting surface using Poisson Disk Sampling

Conclusion and Perspectives

- Methods for building representation of point clouds for 3D object detection:
  - Voxel based
  - Point based
  - Voxel-Point based
  - Perspective: Inference time vs precision
- Central question of Camera-LiDAR fusion - aligning two modalities and methods to solve this
  - Using 3D points from either Point Clouds or Images
  - Using 3D or 2D or both proposals
  - Perspective: Point cloud densification approaches. What about fusion with other modalities and multi-modal dataset?
- Practicality of 3D object detection methods
  - Accuracy-speed tradeoff
  - Robustness to LiDARs' resolution and domain adaptation
  - Perspectives:
    - Efficient hardware design for 3D object detection and new evaluation metrics.
    - Detection with stronger interpretability.
    - Learning 3D detectors from the feedback of planners

Thank you for your attention

Questions?

Acknowledgments to my PhD student Minh-Quan DAO for his contribution to this presentation