

Data, detection and metrics for autonomous vehicles

Oscar Beijbom
Director of Machine Learning, Motional

IPAM Individual Vehicle Autonomy: Perception and Control
October 6 2020



Motional Structure

• **APTIV** •



HYUNDAI
MOTOR GROUP

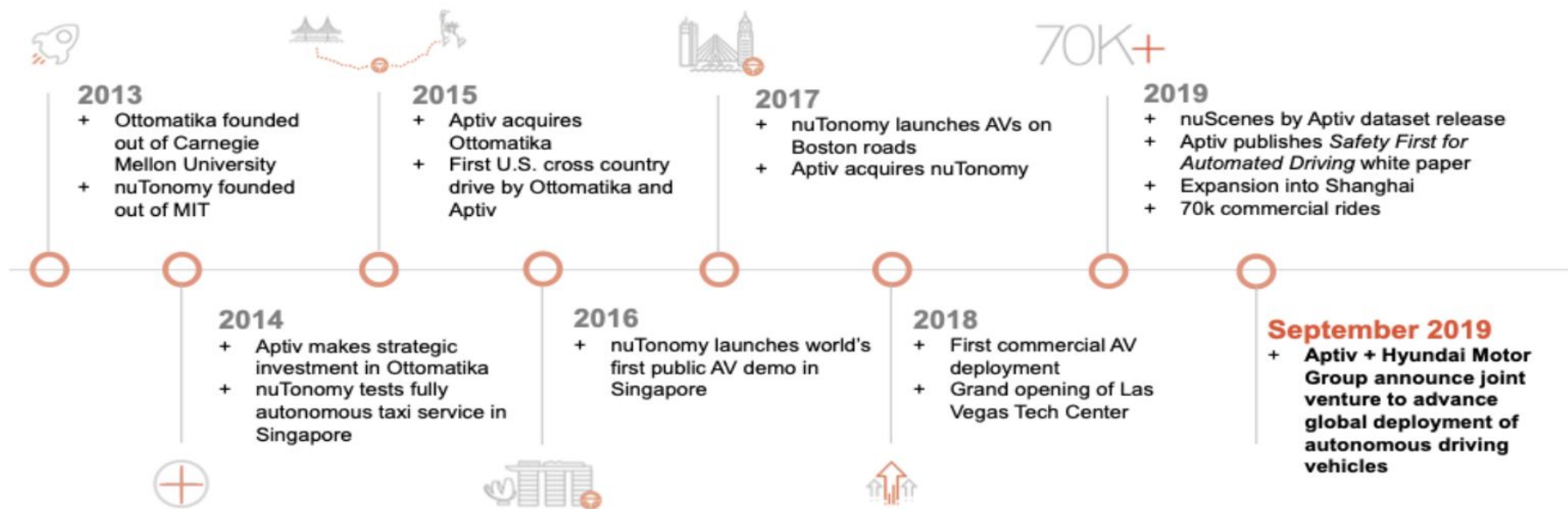
- L4 / L5 automated driving software
- 250+ patents and applications
- ~700 FTEs – including 300 Engineers
- Global footprint (Boston, Pitt, LV, SM, SGP)

- \$1.6bn in cash
- \$400m in-kind contributions:
 - *Vehicle Modification Services; 155 vehicles*
 - *Non-exclusive license to 500+ patents*
 - *70 R&D personnel for 3 years*

Parent Contributions



Our Autonomous Driving Trajectory



Careers in AV

Research & Software

- Planning
- Controls
- Machine Learning
- Localization
- Perception

Program & Product Management

Hardware Engineering

Safety Engineering

Infrastructure Software

- Simulation
- Cybersecurity
- Tools
- DevOps
- Middleware

Validation & Testing

Vehicle & Prototyping

Systems Engineering

Our Offices

Pittsburgh

- Core AV R&D
- Commercialization
- Safety and security
- Vehicle conversion

Las Vegas

- Commercial deployment

Los Angeles

- Machine learning focus
- Core AV R&D

Boston

- Core AV R&D
- Product and marketing
- Safety and security
- Business headquarters

Singapore

- Core AV R&D
- Mobility cloud

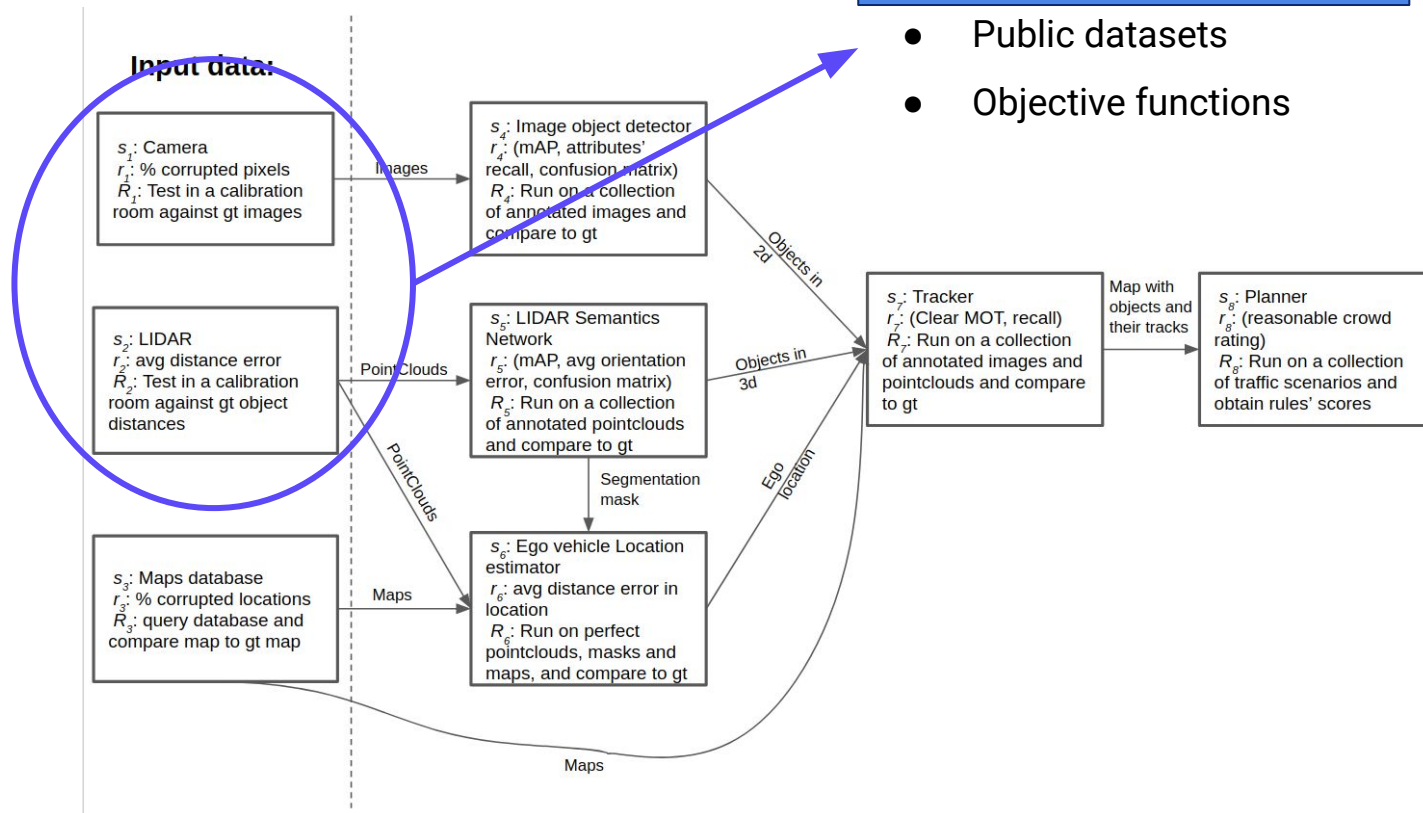
Seoul

- Collaboration with HMC



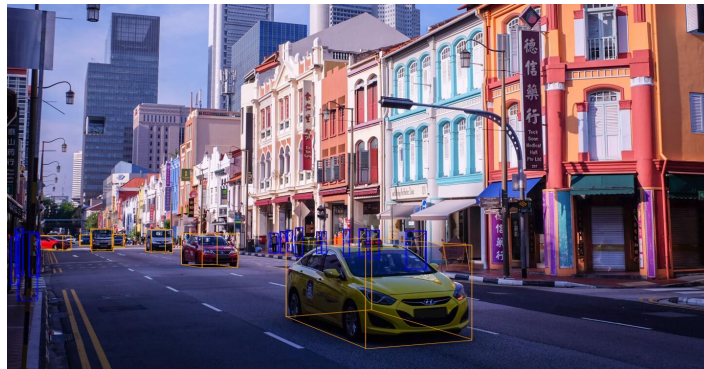
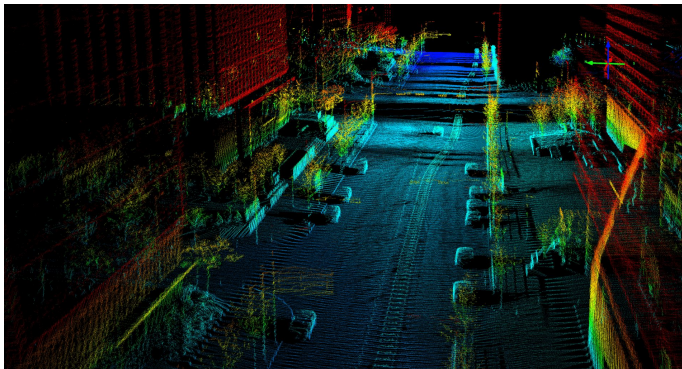
This talk

- Fusion algorithms
- Public datasets
- Objective functions



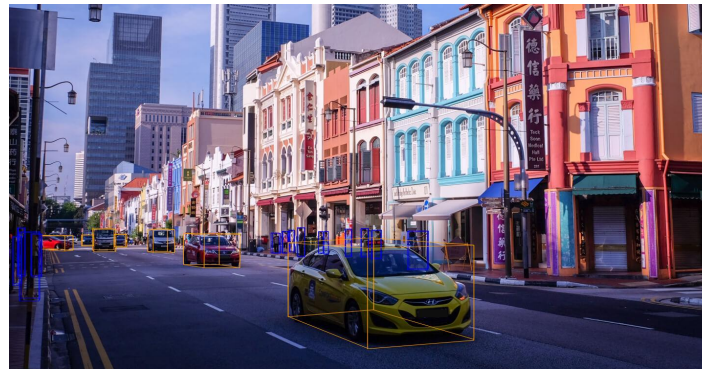
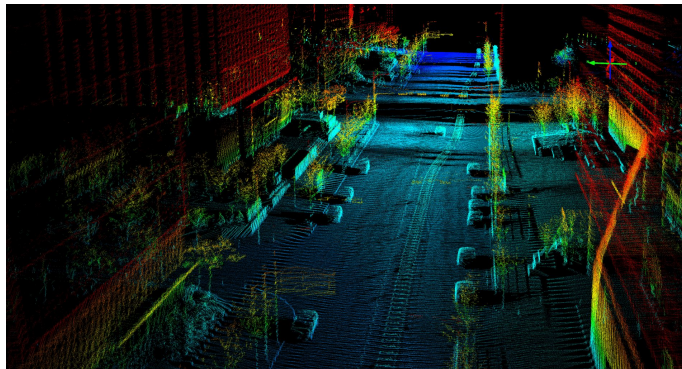
Multi modal object detection

- Input:
 - Image: (nRows, nCols, nChannels) tensor.
 - PointCloud: ((x, y, z, i), nPoints) matrix.
- Output:
 - List of 3d bounding boxes: (**size**, **center**, **orientation**)



Vision and lidar fusion: opportunities

Modality	Range	Shape	Texture	Night	Black surfaces	...
Camera	✗	2D	✓	✗	✓	
Lidar	✓	3D	✗	✓	✗	



Benefits of fusion has been slow to materialize

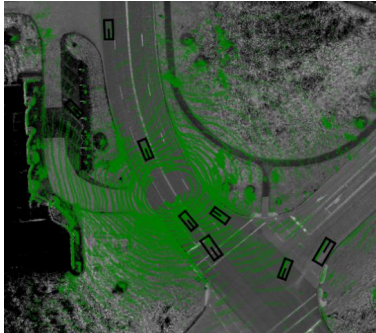
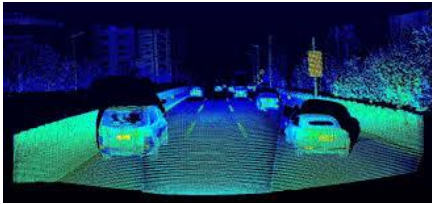

- Lidar only methods outperform the fusion methods on the Kitti benchmark (!)
- Does this mean lidar makes vision redundant for 3D object detection? Surely not!



So why has fusion been so elusive?

One explanation is view-point:

- The 2d conv layer is the workhorse of spatial DL but world is 3d.
 - How to project out data?
 - Front-view or Bird's-eye view?
- Bird's eye view (BEV) dominant:
 - Lack of scale ambiguity.
 - Minimal occlusions.
 - Hard to project images to BEV.
- So what to do?

	Bird's eye view	Front view
Lidar		
Image	<ul style="list-style-type: none">• Structure from motion• Dense depth• ...	

Literature review

Previous fusion methods can be characterized into:

- Front-view fusion
- Object-centric fusion
- Continuous feature fusion
- Transform images to bird's-eye view & perform fusion there
- Use image based 2D detections to seed the 3D detector

Front-view fusion

- Pros
 - Front-view natural for images & point-clouds.
- Cons
 - Depth maps suffer from blurring.
 - Scale and occlusions.
 - Harder to incorporate aux. inputs like map layers.
- Tend not to do well on benchmarks

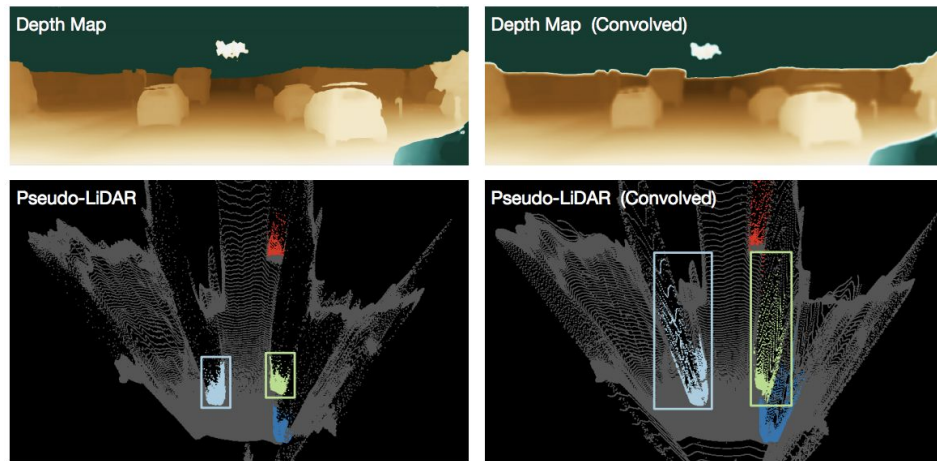
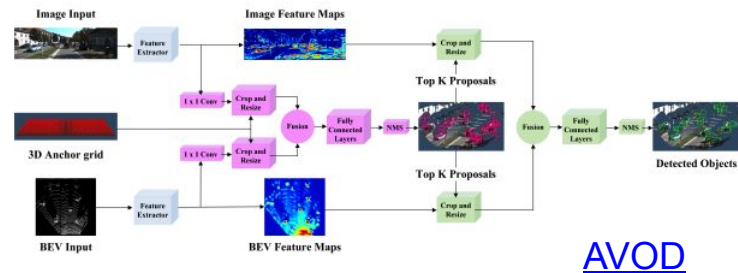
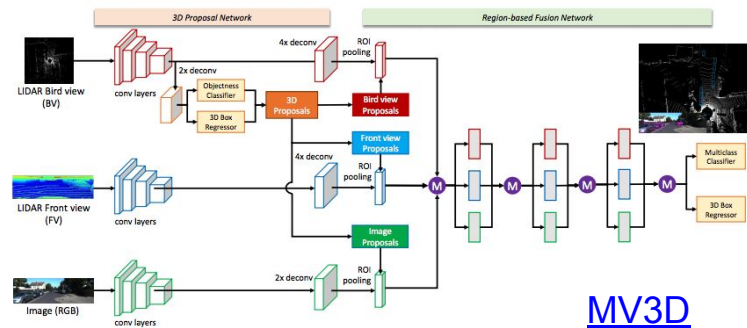


Fig from [PseudoLidar](#)

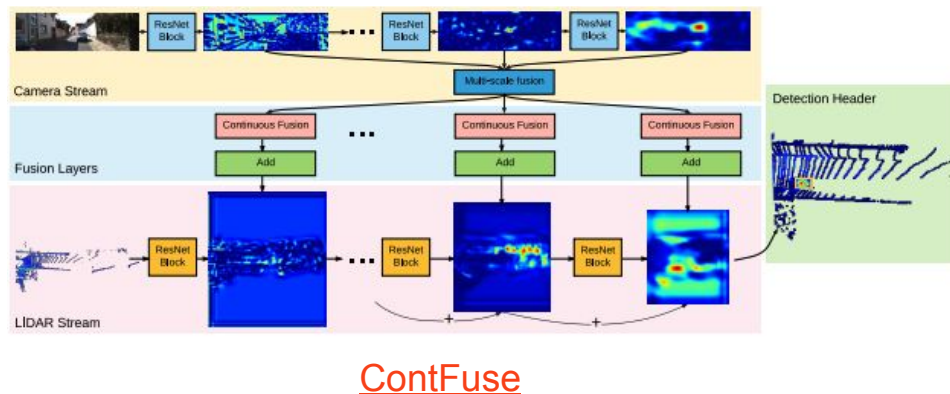
Object Centric Fusion

- Different backbones for FV and BEV.
- Fusion happens at the object proposal level by applying ROI pooling.
- Allows end to end optimization but slow and cumbersome.



Continuous Feature Fusion

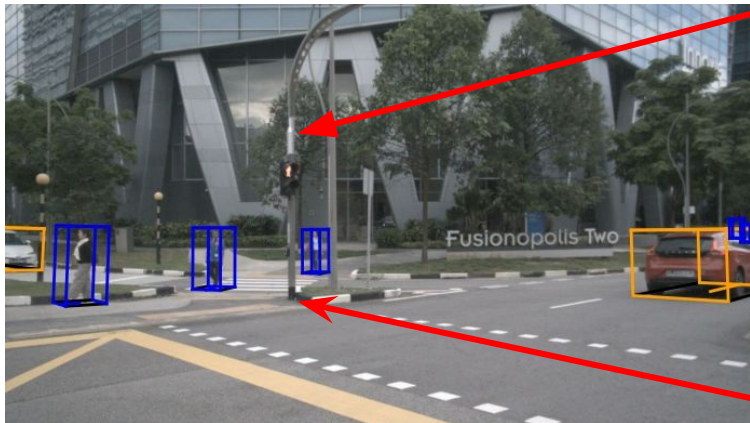
- Allows features to be shared across all strides of image and lidar backbones.
- Drawback: feature blurring.
- ContFuse tries to remedy this based on kNN, bilinear interpolation and a learned MLP but the problem still persists.



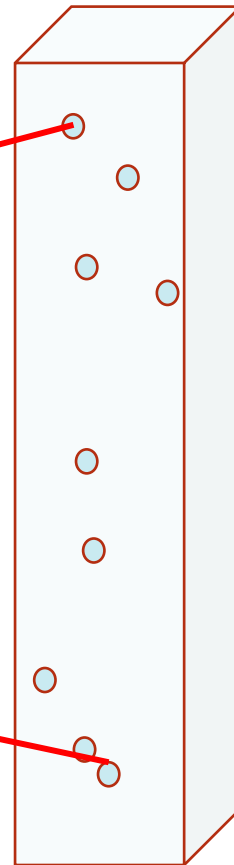
Feature blurring explained

Lidar points in particular x-y ground plane bin

front-view RGB image



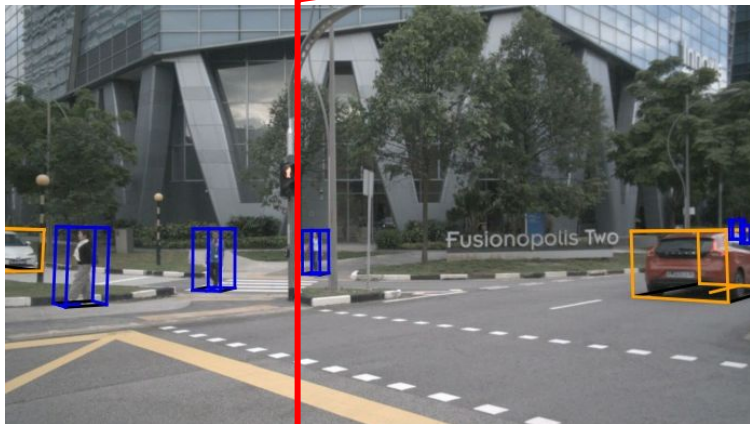
project



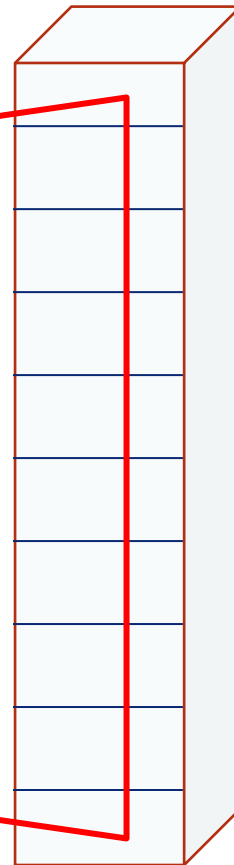
Feature blurring explained

Back bone feature layer

front-view RGB image

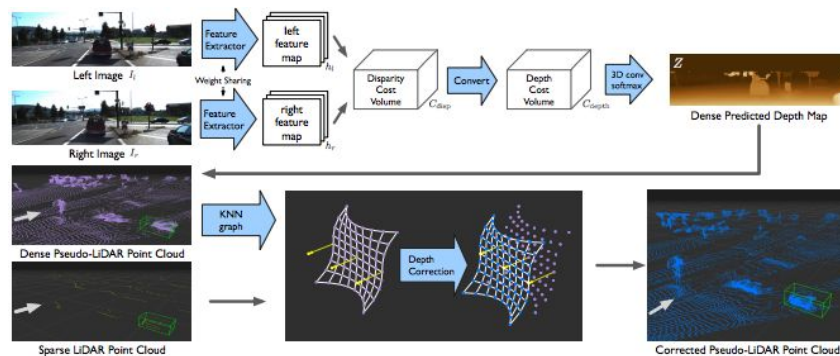


project?



Transform Image to BEV

- Use dense depth estimation to transform images to a BEV representation and do fusion there
- Performance falls short of SOTA and requires several expensive steps of preprocessing to build the pseudo pointcloud.



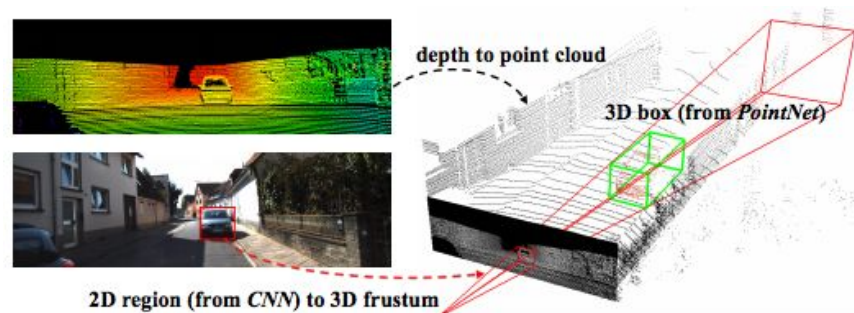
[Pseudo-Lidar++](#)

Using 2D detection seeding

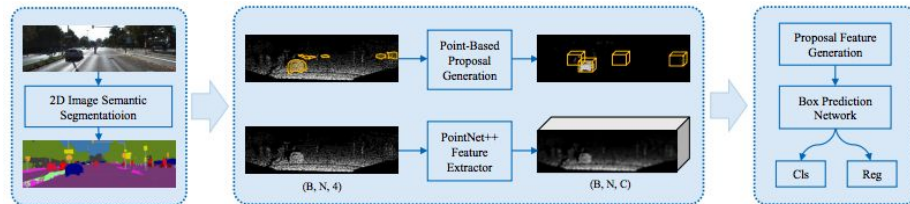
Semantics extracted from image used to seed detection in pointcloud

- Frustum PointNet and ConvNet use 2D detections to limit search space inside the frustum
- IPOD uses semantic segmentation to seed the 3D proposal

Drawback: Imposes an upper bound on recall. Also, computationally expensive.



Frustum PointNet



IPOD

PointPainting: sequential fusion for 3d object detection

- New method for lidar and vision fusion.
- Sequential and combines 3d lidar detectors and image semantic segmentation.
- Improves 3d detection across classes, datasets and detection methods.



PointPainting: Algorithm

Algorithm 1 PointPainting(L, S, T, M)

Inputs:

Lidar point cloud $L \in \mathbb{R}^{N,D}$ with N points and $D \geq 3$.
 Segmentation scores $S \in \mathbb{R}^{W,H,C}$ with C classes.
 Homogenous transformation matrix $T \in \mathbb{R}^{4,4}$.
 Camera matrix $M \in \mathbb{R}^{3,4}$.

Output:

Painted lidar points $P \in \mathbb{R}^{N,D+C}$

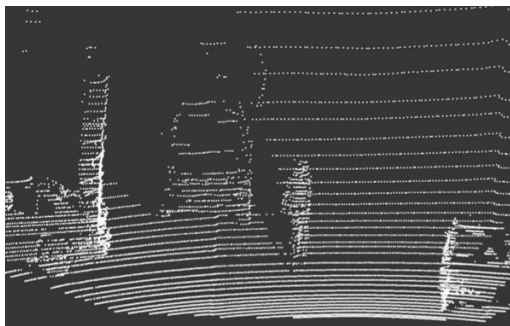
for $\vec{l} \in L$ **do**

$\vec{l}_{\text{image}} = \text{PROJECT}(M, T, \vec{l}_{xyz})$ $\triangleright \vec{l}_{\text{image}} \in \mathbb{R}^2$

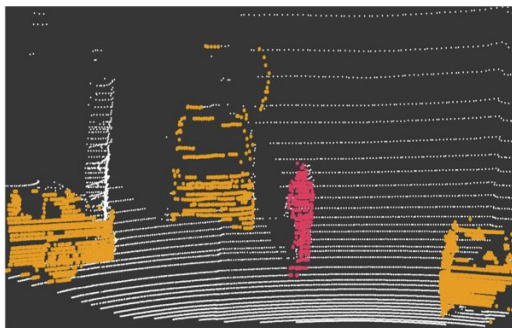
$\vec{s} = S[\vec{l}_{\text{image}}[0], \vec{l}_{\text{image}}[1], :]$ $\triangleright \vec{s} \in \mathbb{R}^C$

$\vec{p} = \text{Concatenate}(\vec{l}, \vec{s})$ $\triangleright \vec{p} \in \mathbb{R}^{D+C}$

end for

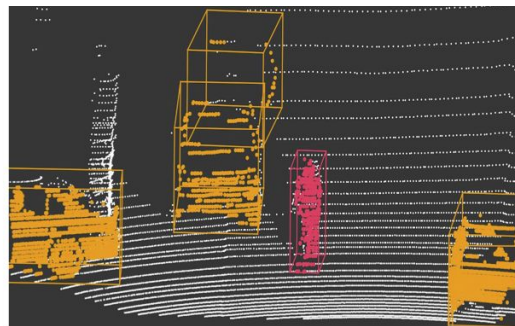


2
Point
Painting



3
Lidar
Detector

e.g.
Point-RCNN
PointPillars
etc



2 Point Painting



1
Sem. Seg



Comparison with literature

Addresses the shortcomings of the previous methods

- Does not add any restriction on the 3D detection architecture
- Does not suffer from feature or depth blurring
- Does not require a pseudo-pointcloud to be computed
- Does not limit the maximum recall

KITTI Results (val set)

Method	mAP	Car			Pedestrian			Cyclist		
	Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointPillars [11]	73.78	90.09	87.57	86.03	71.97	67.84	62.41	85.74	65.92	62.40
Painted PointPillars	76.27	90.01	87.65	85.56	77.25	72.41	67.53	81.72	68.76	63.99
Delta	+2.50	-0.08	0.08	-0.47	+5.28	+4.57	+5.12	-4.02	+2.84	+1.59
VoxelNet [34, 29]	71.83	89.87	87.29	86.30	70.08	62.44	55.02	85.48	65.77	58.97
Painted VoxelNet	73.55	90.05	87.51	86.66	73.16	65.05	57.33	87.46	68.08	65.59
Delta	+1.71	+0.18	+0.22	+0.36	+3.08	+2.61	+2.31	+1.98	+2.31	+6.62
PointRCNN [21]	72.42	89.78	86.19	85.02	68.37	63.49	57.89	84.65	67.59	63.06
Painted PointRCNN	75.80	90.19	87.64	86.71	72.65	66.06	61.24	86.33	73.69	70.17
Delta	+3.37	+0.41	+1.45	+1.69	+4.28	+2.57	+3.35	+1.68	+6.10	+7.11

Table 1. PointPainting applied to state of the art lidar based object detectors. All lidar methods show an improvement in bird's-eye view (BEV) mean average precision (mAP) of car, pedestrian, and cyclist on KITTI *val* set, moderate split.

KITTI Results (test set)

Method	Modality	mAP	Car			Pedestrian			Cyclist		
		Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
MV3D[3]	L & I	N/A	86.62	78.93	69.80	N/A	N/A	N/A	N/A	N/A	N/A
AVOD-FPN[9]	L & I	64.07	90.99	84.82	79.62	58.49	50.32	46.98	69.39	57.12	51.09
IPOD[30]	L & I	64.60	89.64	84.62	79.96	60.88	49.79	45.43	78.19	59.40	51.38
F-PointNet[17]	L & I	65.20	91.17	84.67	74.77	57.13	49.57	45.48	77.26	61.37	53.78
F-ConvNet[25]	L & I	67.89	91.51	85.84	76.11	57.04	48.96	44.33	84.16	68.88	60.05
MMF[12]	L, I & M	N/A	93.67	88.21	81.99	N/A	N/A	N/A	N/A	N/A	N/A
LaserNet[16]	L	N/A	79.19	74.52	68.45	N/A	N/A	N/A	N/A	N/A	N/A
SECOND[28]	L	61.61	89.39	83.77	78.59	55.99	45.02	40.93	76.5	56.05	49.45
PointPillars[10]	L	65.98	90.07	86.56	82.81	57.60	48.64	45.78	79.90	62.73	55.58
STD[31]	L	68.38	94.74	89.19	86.42	60.02	48.72	44.55	81.36	67.23	59.35
PointRCNN[20]	L	66.92	92.13	87.39	82.72	54.77	46.13	42.84	82.56	67.24	60.28
Painted PointRCNN	L & I	69.86	92.45	88.11	83.36	58.70	49.93	46.29	83.91	71.54	62.97
Delta	ΔI	+2.94	+0.32	+0.72	+0.64	+3.93	+3.80	+3.45	+1.35	+4.30	+2.69

Table 2. Results on the KITTI test BEV detection benchmark. The modalities are lidar (L), images (I), and maps (M). The delta is the difference due to Painting, ie Painted PointRCNN minus PointRCNN.

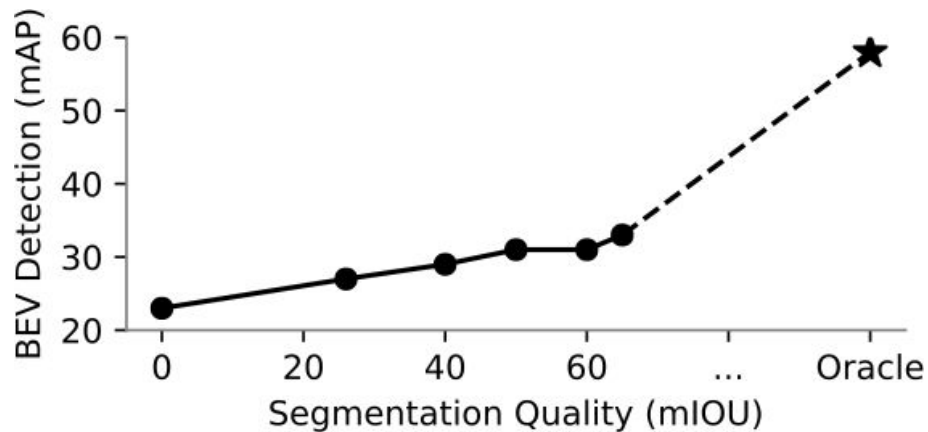
nuScenes Results (test set)

Methods	mAP	Car	Truck	Bus	Trailer	Ctr. Vhl.	Ped.	Motorcycle	Bicycle	Tr. Cone	Barrier
PointPillars [10, 1]	30.5	68.4	23.0	28.2	23.4	4.1	59.7	27.4	1.1	30.8	38.9
PointPillars+	40.1	76.0	31.0	32.1	36.6	11.3	64.0	34.2	14.0	45.6	56.4
Painted PointPillars+	46.4	77.9	35.8	36.1	37.3	15.8	73.3	41.5	24.1	62.4	60.2
Delta	+6.3	+1.9	+4.8	+3.9	+0.7	+4.5	+9.3	+7.3	+10.1	+16.8	+3.8

Table 3. Per class nuScenes performance. Evaluation of detections as measured by average precision (AP) or mean AP (mAP) on nuScenes test set. Abbreviations: construction vehicle (Ctr. Vhl.), pedestrian (Ped.), and traffic cone (Tr. Cone).

PointPainting: Ablation Study

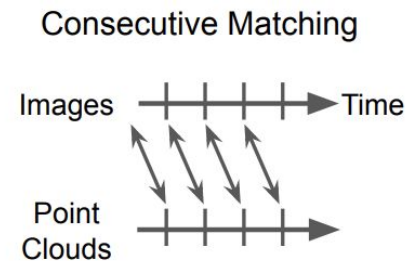
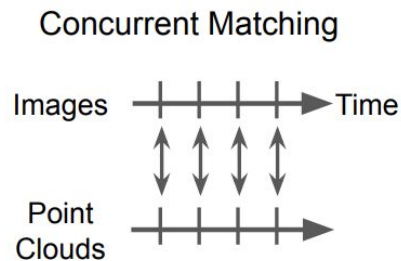
- Better image based semantic segmentation model => better 3D results from PointPainting
- Oracle: Use GT 3D boxes to paint all points (instead of using predicted semantic segmentation scores). Used to simulate perfect semantic segmentation.



PointPainting: Latency Analysis

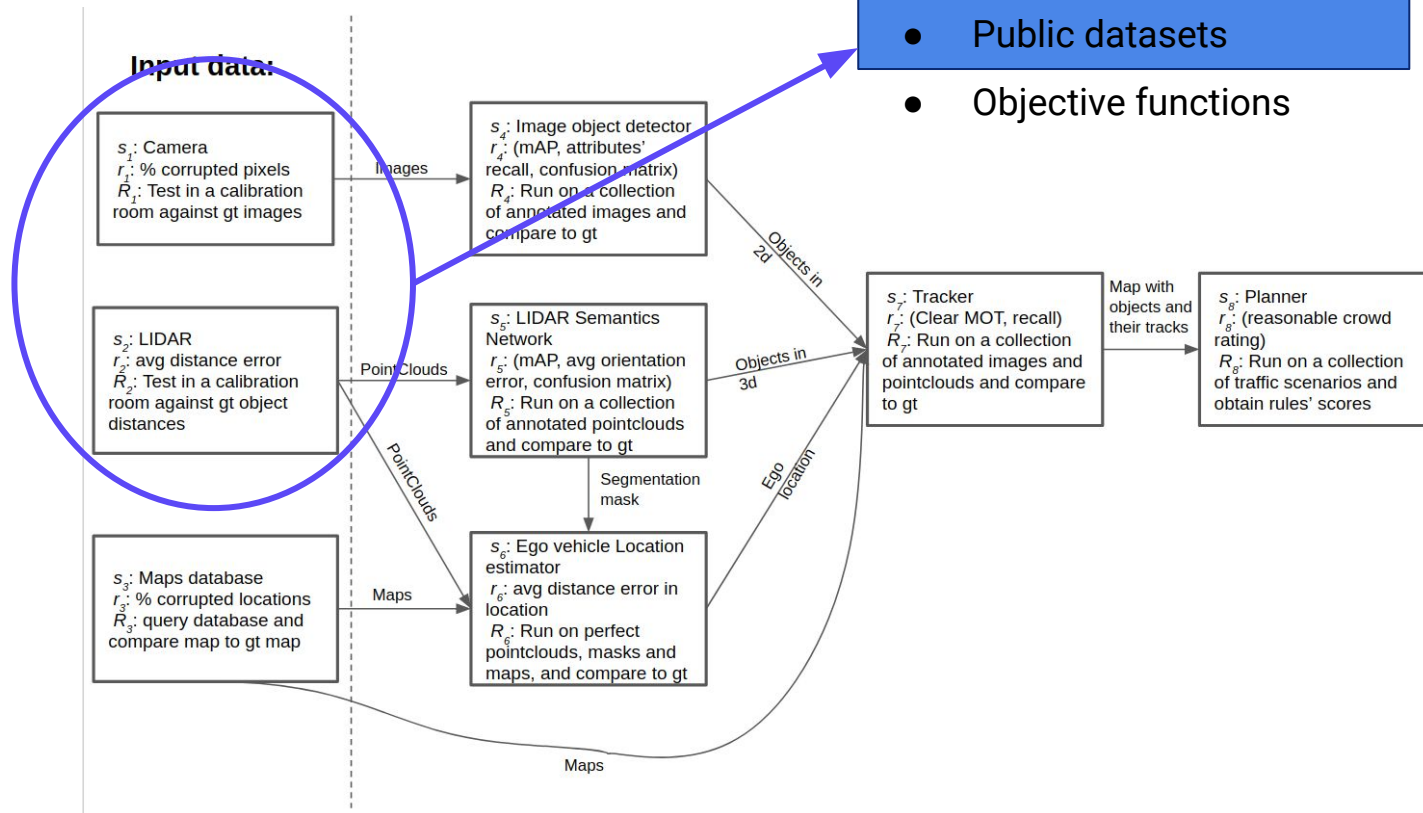
- Latency can be minimized by pipelining thereby making the runtimes of ‘painted’ methods similar to its lidar only baseline.

Method	Matching	NDS	mAP	Latency
Painted PointPillars	Concurrent	46.3	33.9	Time (PointPillars) + Time (Img. Seg.)
Painted PointPillars	Consecutive	46.4	33.9	Time (PointPillars)



This talk

- Fusion algorithms
- Public datasets
- Objective functions



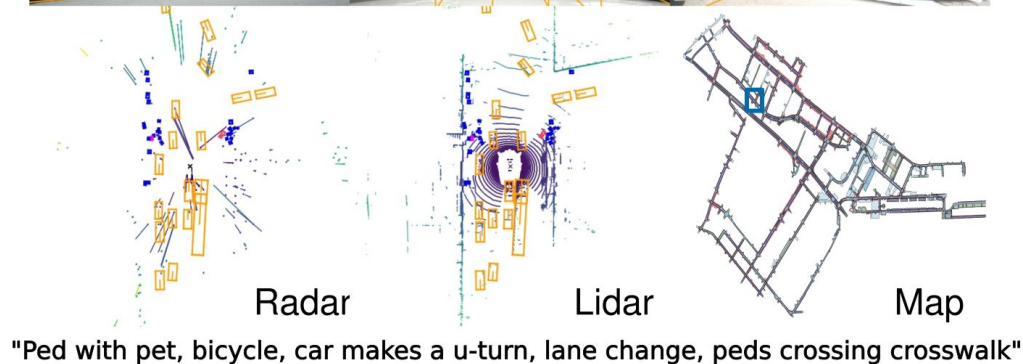
(Non-public) datasets

- Building data-engines a core part of applied ML.
- Most critical and nerve wracking part of building a ML stack!
- How to mine for the right data?
- How to annotate large amounts of data cheaply?
- How to define the right taxonomy?

Good news is that we have done the hard work
for you!

nuScenes: a multimodal dataset for autonomous driving

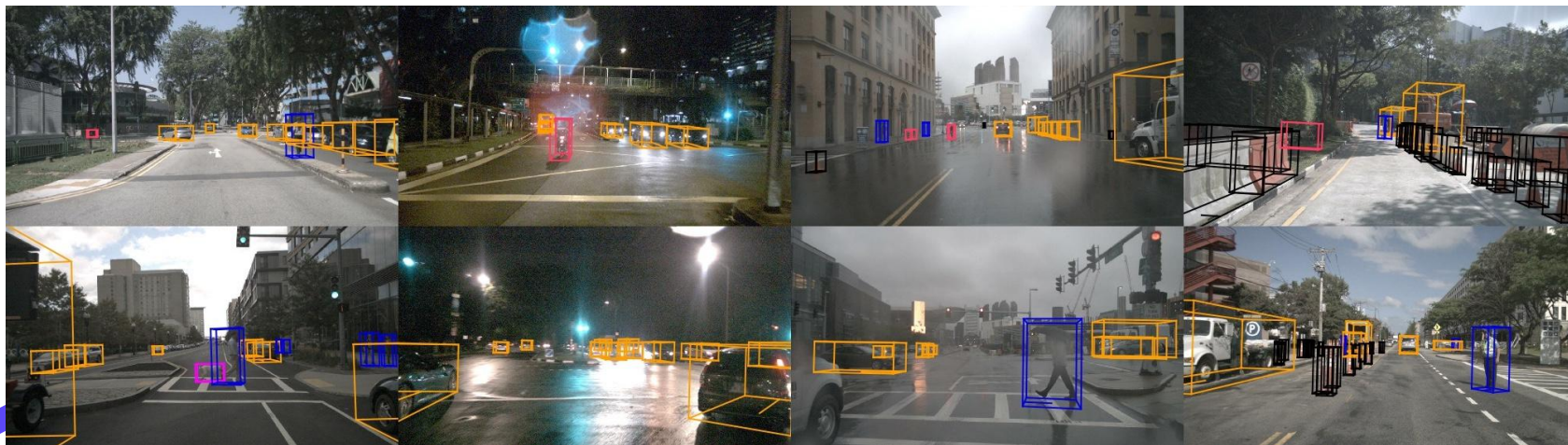
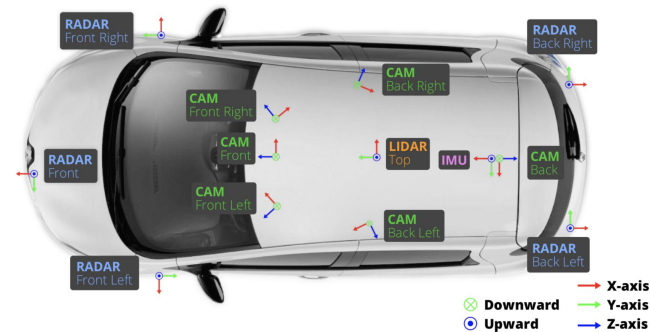
- 1000 20-second scenes
- Synced sensors w/ 360 view
- High-def maps
- Fully annotated in 3D
- Free for research



<https://www.nuscenes.org/>

nuScenes: diversity

- Interesting maneuvers and rare classes
- 4 diverse locations in Boston and Singapore
- Left-hand vs. right-hand driving
- Different vehicle and vegetation types
- Night-time and rainy data



nuScenes: friends and followers

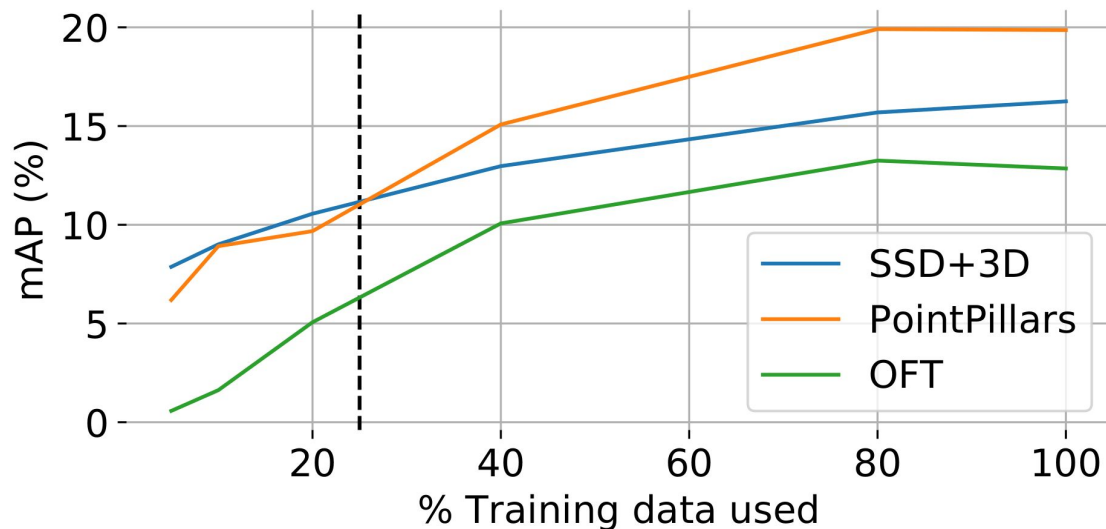
Dataset	Year	Scenes	Size (hr)	RGB imgs	PCs lidar ^{††}	PCs radar	Ann. frames	3D boxes	Night / Rain	Map layers	Classes	Locations
CamVid [8]	2008	4	0.4	18k	0	0	700	0	No/No	0	32	Cambridge
Cityscapes [19]	2016	n/a	-	25k	0	0	25k	0	No/No	0	30	50 cities
Vistas [33]	2017	n/a	-	25k	0	0	25k	0	Yes/Yes	0	152	Global
BDD100K [85]	2017	100k	1k	100M	0	0	100k	0	Yes/Yes	0	10	NY, SF
ApolloScope [41]	2018	-	100	144k	0**	0	144k	70k	Yes/No	0	8-35	4x China
D ² -City [11]	2019	1k [†]	-	700k [†]	0	0	700k [†]	0	No/Yes	0	12	5x China
KITTI [32]	2012	22	1.5	15k	15k	0	15k	200k	No/No	0	8	Karlsruhe
AS lidar [54]	2018	-	2	0	20k	0	20k	475k	-/-	0	6	China
KAIST [17]	2018	-	-	8.9k	8.9k	0	8.9k	0	Yes/No	0	3	Seoul
H3D [61]	2019	160	0.77	83k	27k	0	27k	1.1M	No/No	0	8	SF
nuScenes	2019	1k	5.5	1.4M	400k	1.3M	40k	1.4M	Yes/Yes	11	23	Boston, SG
Argoverse [10]	2019	113 [†]	0.6 [†]	490k [†]	44k	0	22k [†]	993k [†]	Yes/Yes	2	15	Miami, PT
Lyft L5 [45]	2019	366	2.5	323k	46k	0	46k	1.3M	No/No	7	9	Palo Alto
Waymo Open [76]	2019	1k	5.5	1M	200k	0	200k[‡]	12M[‡]	Yes/Yes	0	4	3x USA
A*3D [62]	2019	n/a	55	39k	39k	0	39k	230k	Yes/Yes	0	7	SG
A2D2 [34]	2019	n/a	-	-	-	0	12k	-	-/-	0	14	3x Germany

Table 1. AV dataset comparison. The top part of the table indicates datasets without range data. The middle and lower parts indicate datasets (not publications) with range data released until and after the initial release of this dataset. We use bold highlights to indicate the best entries in every column among the datasets with range data. Only datasets which provide annotations for at least *car*, *pedestrian* and *bicycle* are included in this comparison. (†) We report numbers only for scenes annotated with cuboids. (‡) The current Waymo Open dataset size is comparable to nuScenes, but at a 5x higher annotation frequency. (††) Lidar pointcloud count collected from *each lidar*. (**) [41] provides static depth maps. (-) indicates that no information is provided. SG: Singapore, NY: New York, SF: San Francisco, PT: Pittsburgh, AS: ApolloScope.

nuScenes: experiments

- Larger datasets are needed

70% relative improvement with Pointpillars vs. KITTI amounts of training data



nuScenes: experiments

- Multiple lidar sweeps drastically improve performance
- Pre-training on KITTI / ImageNet only gives a small improvement

Lidar sweeps	Pretraining	NDS (%)	mAP (%)	mAVE (m/s)
1	KITTI	31.8	21.9	1.21
5	KITTI	42.9	27.7	0.34
10	KITTI	44.8	28.8	0.30
10	ImageNet	44.9	28.9	0.31
10	None	44.2	27.6	0.33

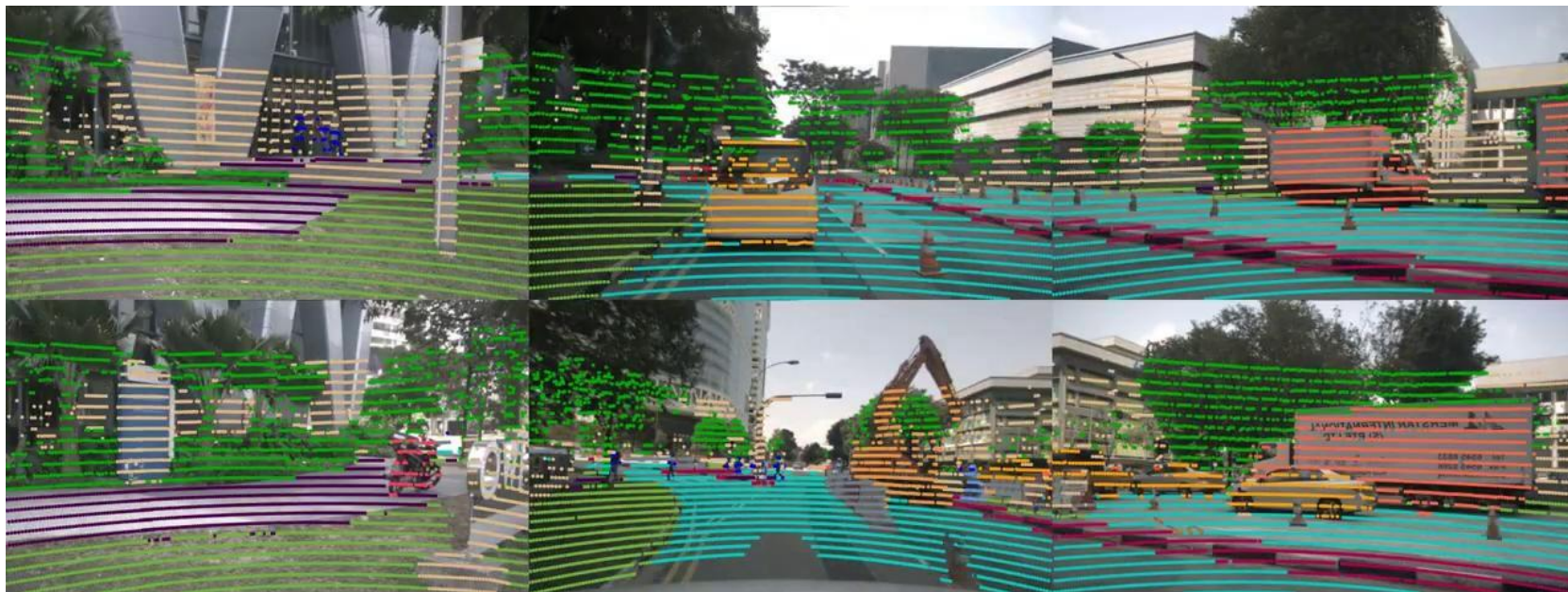
Table 3. PointPillars [51] detection performance on the val set. We can see that more lidar sweeps lead to a significant performance increase and that pretraining with ImageNet is on par with KITTI.

nuScenes: expansions

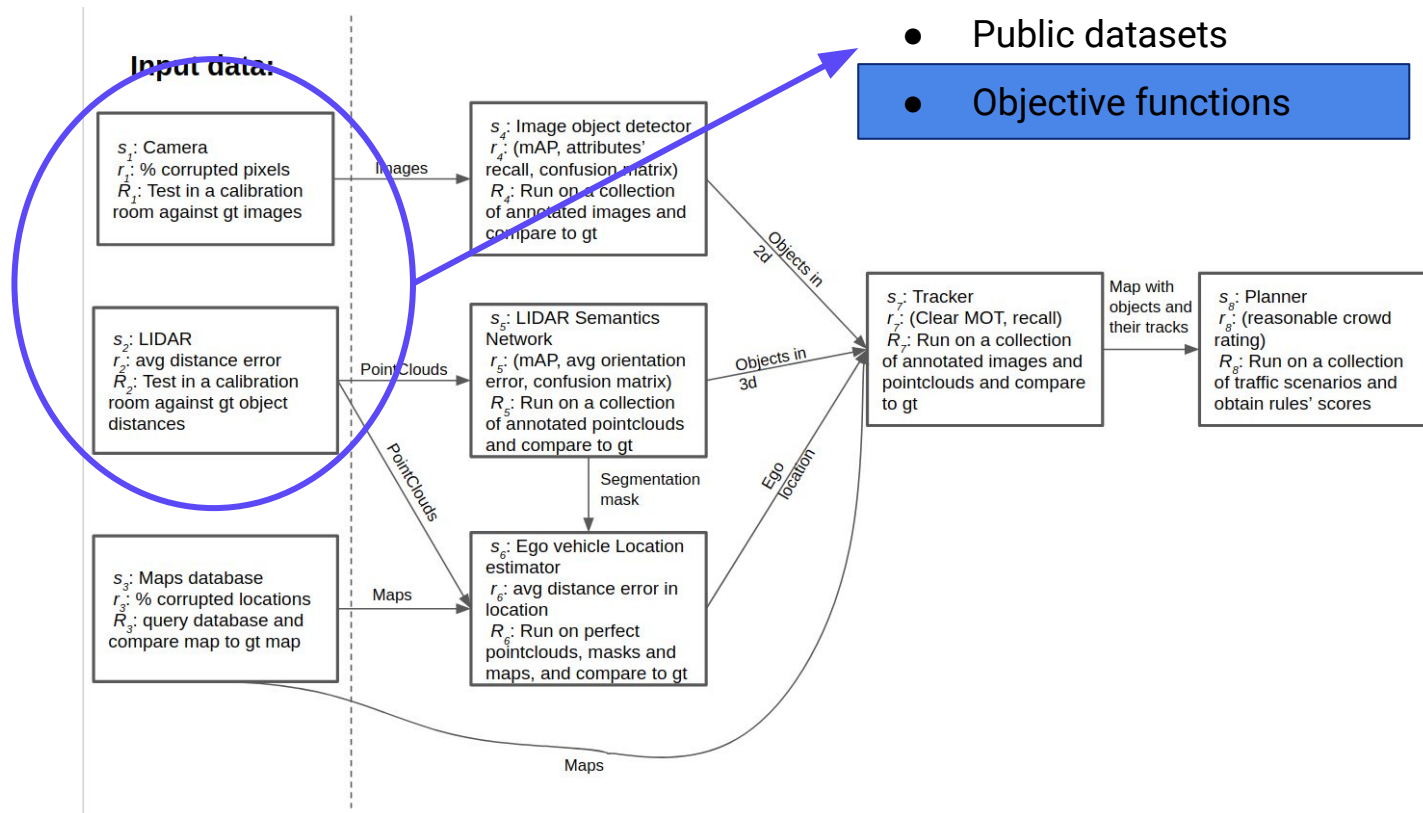
- **nuImages**
 - 100k images with 800k 2d boxes and masks
 - Depth maps
 - Temporal images and ego-poses
- **nuScenes-lidarseg**
 - 40k keyframes with point-level labels for 1.1 Billion points
 - Go beyond bounding boxes; focus on stuff classes (road, sidewalk, building)



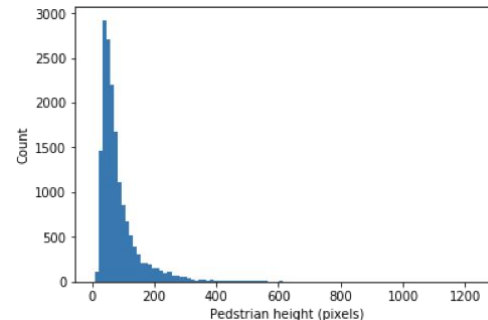
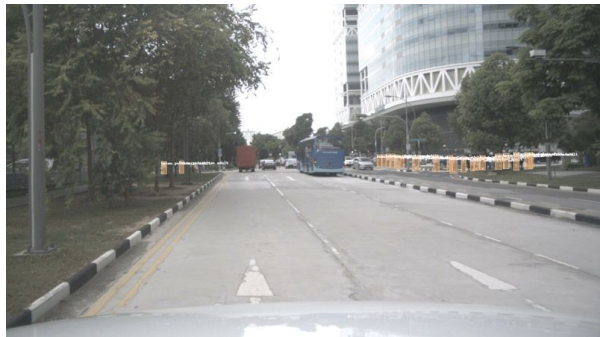
nuScenes lidarseg



This talk



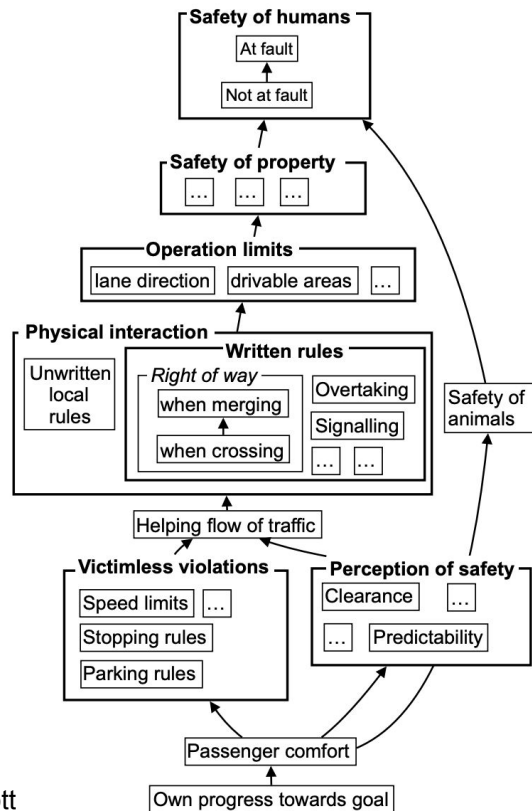
NN's will learn what you ask them to learn: no more, no less



Prototype image detector had solid mAP & did well on small objects, but missed nearby objects!

The AV objective function

- Need function $f(w)$: scenario \rightarrow score.
 - To optimize our stack
 - For retrospection if incidents happen.
- Not obvious what f should look like!
 - Legal, ethics, culture all impose constraints.
 - Often conflicting.
- The “Rulebooks” idea address this.
 - Each aspect encoded as a “rule”
 - Partial ordering across rules.



Rule R3a: Stay in drivable area

Variables (in addition to those from A1):

$d_{\text{tot, left}}$: maximum infringement of the ego to the left boundary of the drivable area

$d_{\text{tot, right}}$: maximum infringement of the ego to the right boundary of the drivable area

Parameters:

W_{road} : road width multiplied by a coefficient

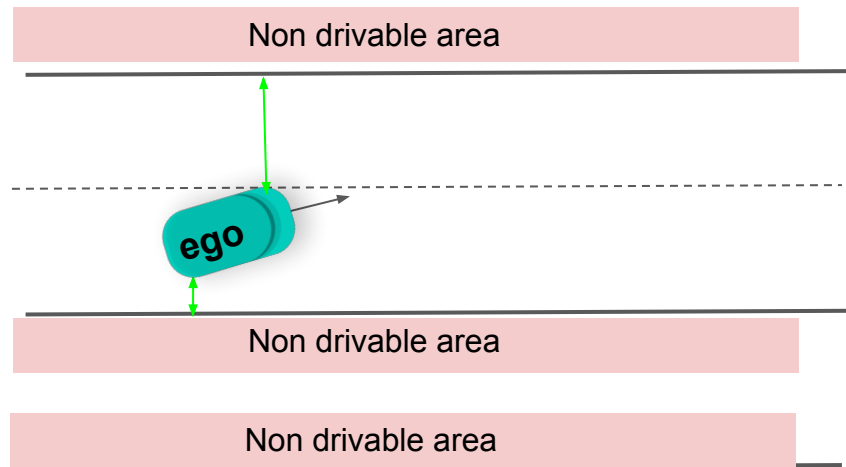
Violation metric at each time:

$$\varrho(t) = (d_{\text{tot, left}}(t) + d_{\text{tot, right}}(t)) / W_{\text{road}}$$

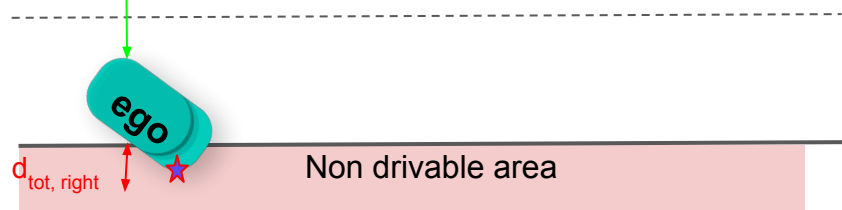
Violation metric over time for each instance:

$$\varrho = (1/T) \int_{[0, T]} \varrho(t)$$

Rule satisfied



Rule violated



But how to combine the rules?

“So what would a lawyer say about all of this?”

“An AV would need to internalize what a reasonable person would do.”

-- Emilio Frazzoli’

The reasonable crowd

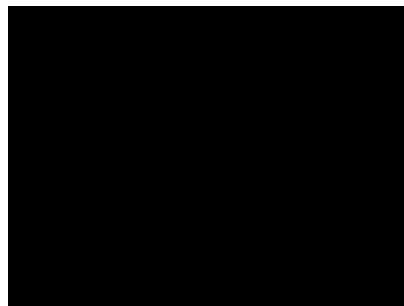
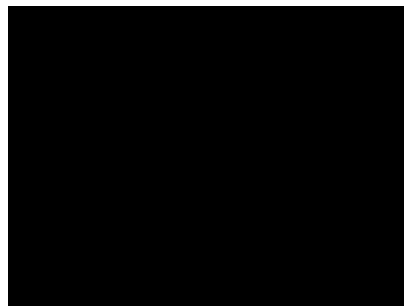
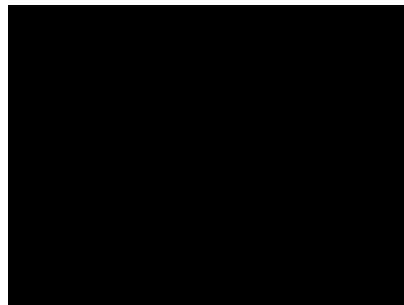
- Show pairs of videos and ask what a “reasonable” driver would do.
- Learn a mapping from rules to human preferences.
- $f(w)$ score of scenario w .
 - $f(w) = f_{\text{crowd}}(f_{\text{rules}}(w))$
 - $f_{\text{rules}}(w)$ is an explicit rulebooks encoding
 - f_{crowd} is learned from data.

Use data to model complex relationship. Linear function allows inspection



Pilot study

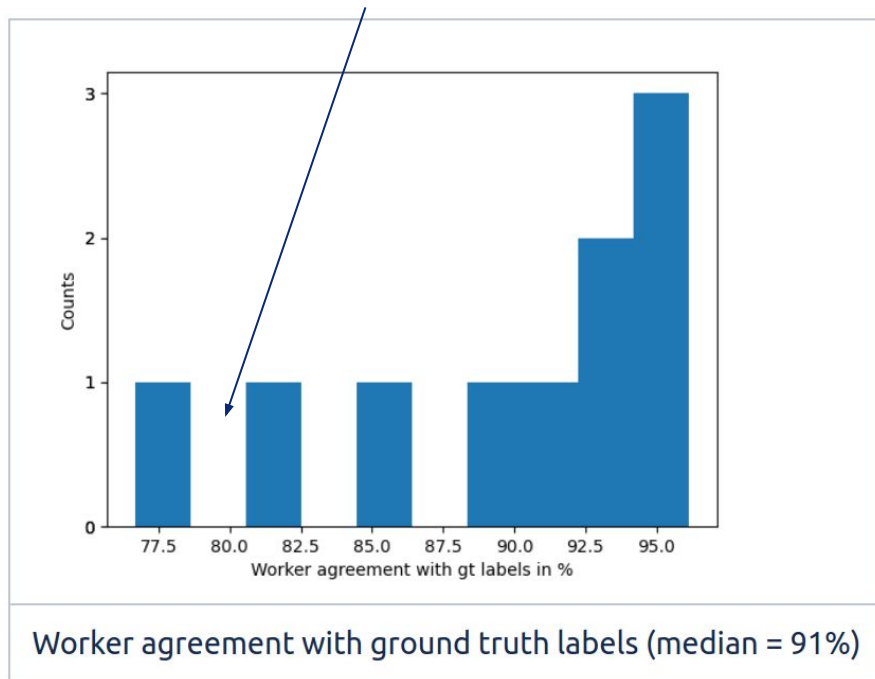
- 30 scenarios (map + agents).
- A total of 147 trajectories & 376 unique trajectory pairs.
- Median trajectory length is 9s. There is a total of ~ 24 mins of driving time.
- Each trajectory pair is annotated 6 times.
- No stop signs or traffic lights



Insight #1: Workers agree and a linear model can model this

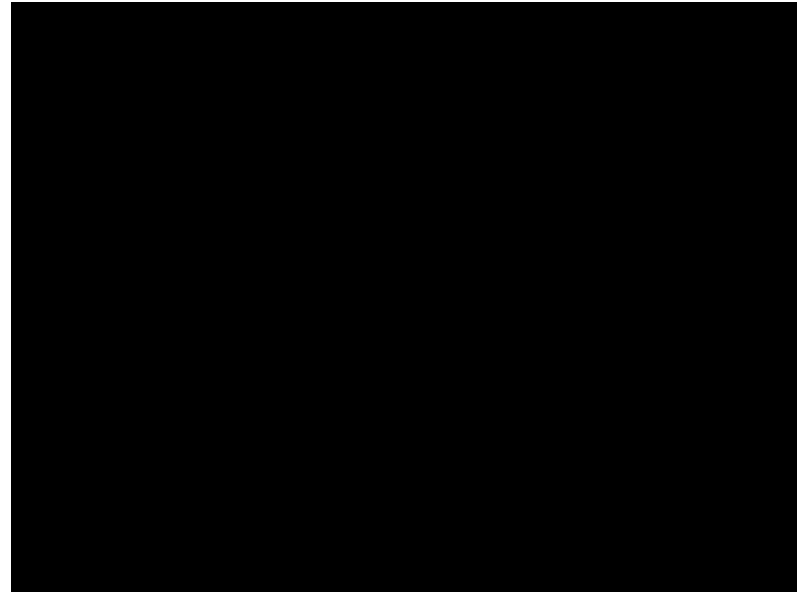
Model name	test accuracy
Logistic regression	81.4%
Linear SVM	80.5%
rbf SVM	79.7%
MLP	77.9%
Random Forest	72%

Our best model would be
2nd worst worker :P



Insight #2: Example rankings for Logistic Regression (LR)

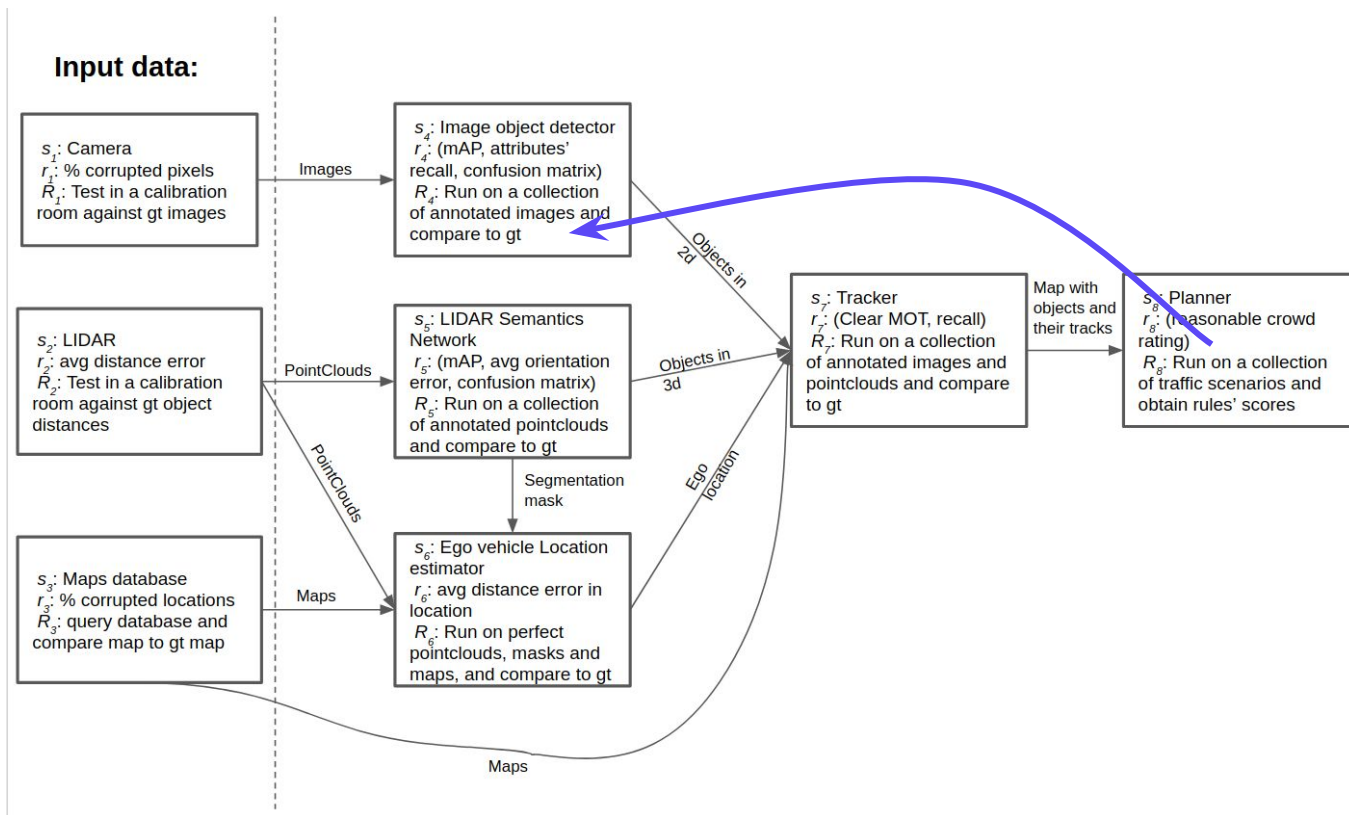
Track id	% preferred	LR rank
19-1-5	100	1
19-1-2	77	2
19-1-3	70	3
19-1-4	44	4
19-1-1	41	5
19-1-7	13	6



Pilot 3 ML results: Importance of different rules

Rule Name	Logistic Regression Weight
No Collision	16.1%
Pedestrian clearance off road	14.9%
Parked car clearance	13.1%
Pedestrian clearance on road	11.9%
Stay on drivable area	10.8%
Stay in lane	10%
Crosswalk with vulnerable road users	9.5%
Max speed	8.8%
Drive smoothly	5%

Mapping back to sub-systems (ongoing work: let's chat later)



Thanks to all co-authors and colleagues!

Interested in a position?

[https://motional.com/careers/
lisa.kattan@motional.com](https://motional.com/careers/lisa.kattan@motional.com)

Get started with R&D?

www.nuscenes.org

