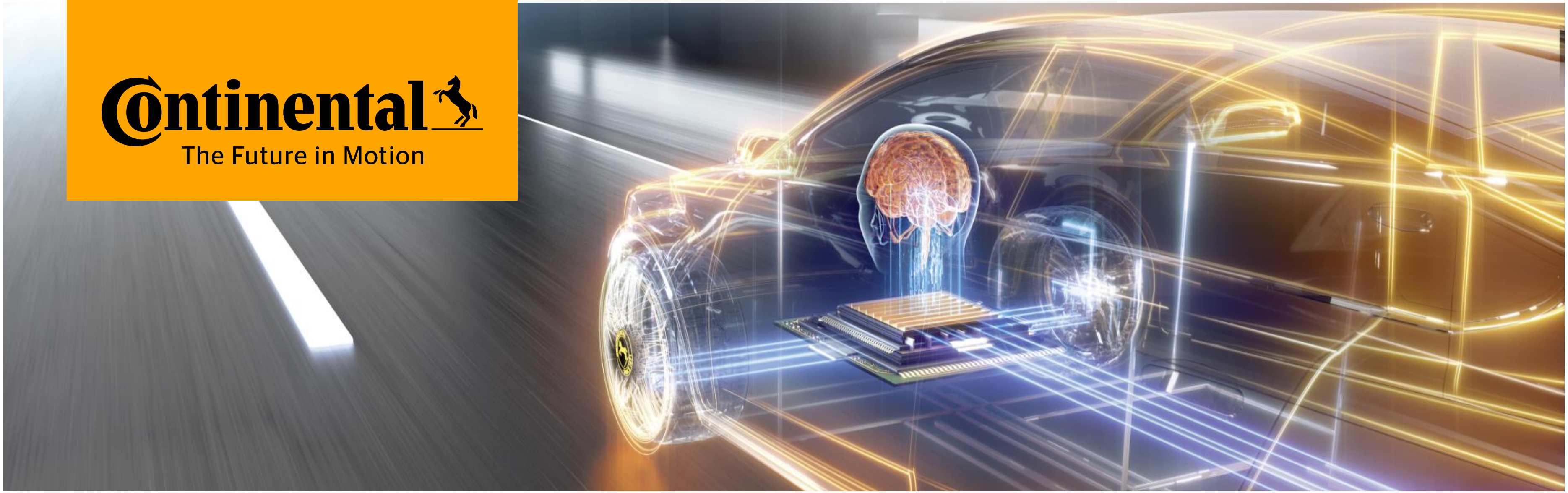
Continental
The Future in Motion



Playing detective
Dissecting silent failures of Deep Learning models for 3D Point Clouds

Robert-Zsolt Kabai
Team Lead, Deep Learning for Point Cloud
Machine Learning Methods
Deep Learning Competence Center - Continental

Agenda

Continental ADAS Products

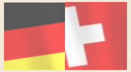
History

- Continental is much more than tires!

Continental ADAS Products

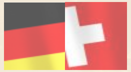
History

- Continental is much more than tires!

1996	1999	2005	2013
			
Foundation of ADAS	Adaptive Cruise Control (*Daimler*)	First Camera (*Volvo*)	1st Multifunction Stereo Camera (*Daimler*)

Continental ADAS Products
History

- Continental is much more than tires!

1996	1999	2005	2013
			
Foundation of ADAS	Adaptive Cruise Control (*Daimler*)	First Camera (*Volvo*)	1st Multifunction Stereo Camera (*Daimler*)

Today:

- Millions of sensors sold
- Many Development centers worldwide
- Competence Center Deep Machine Learning

On the Way to Automated Driving

NCAP
New Car Assessment Programme

EURO NCAP
JNCAP
LATIN NCAP
ASEAN NCAP

1

Automated Parking

2

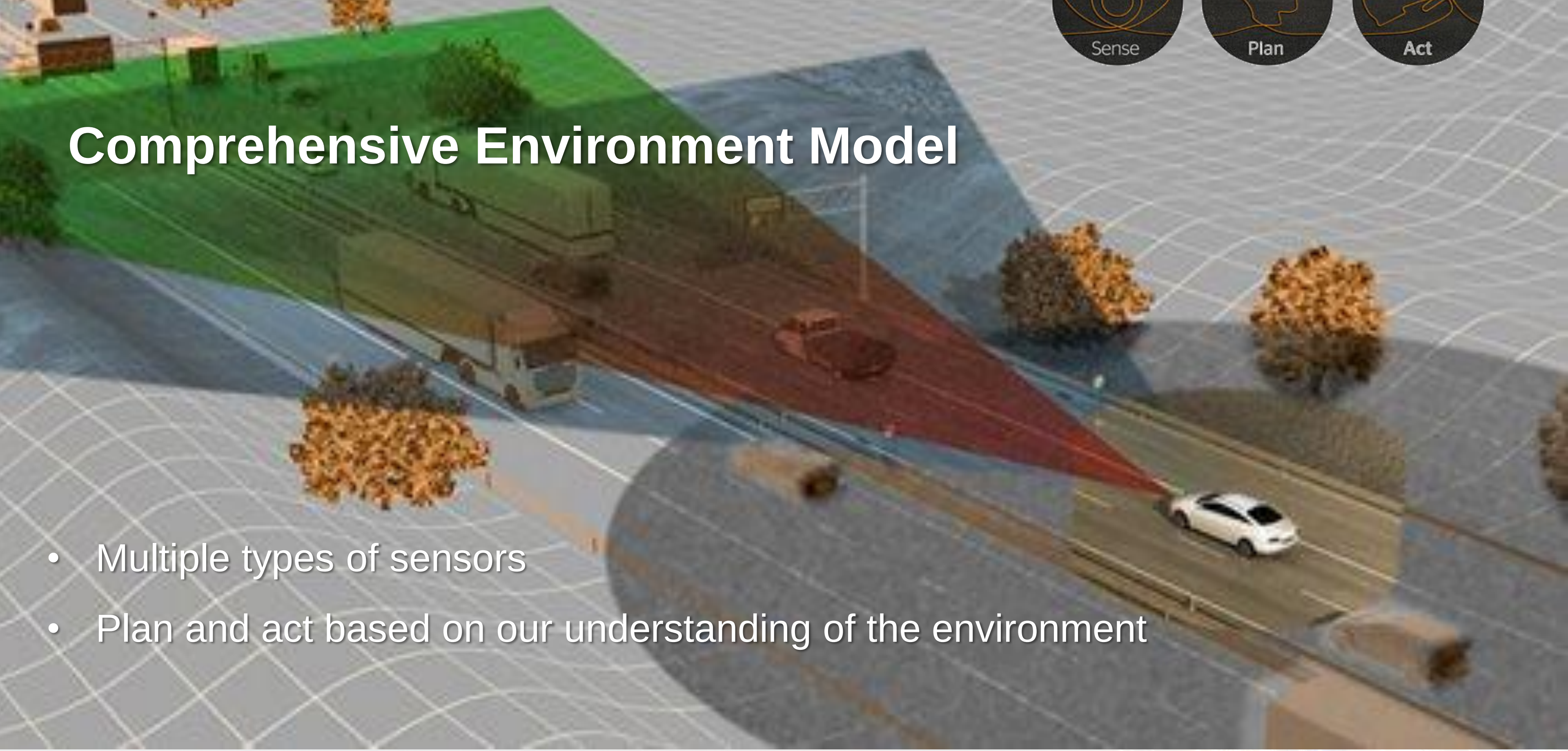
Partly / Highly Automated Driving

3

Self Driving Car

4

Comprehensive Environment Model



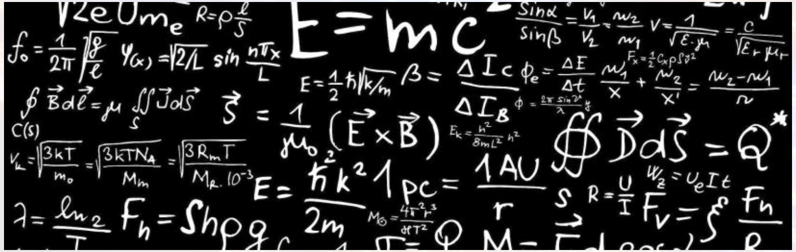
- Multiple types of sensors
- Plan and act based on our understanding of the environment

ADAS Fusion & Perception
What is CEM?

Comprehensive Environment Model

Real world Deep Learning challenges

- The real world is often too complicated to model it completely
- Deep Learning is an approach to deal with that



Popular 3D Data Representation Methods

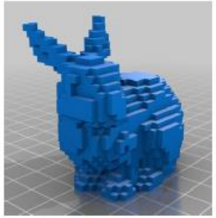
No standard approach for working with 3D

- E.g. for 2-D computer vision the de facto „to-go“ approach are ConvNets.

2D projection

Learn 2D image depth map.

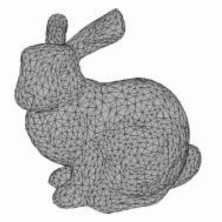
Fallback to 2D methods.



Volumetric

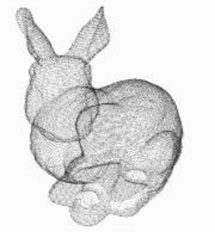
Voxelize and use 3D convolutions.

Marrying 2D methods to approximate 3D.



Mesh

Set of points with structural information.



Point cloud

Use real 3D data as is.

True 3D, but hard problem.

Point Cloud challenges

Varying amounts of detail seen in real life point clouds

Impressive

Point Cloud challenges

Varying amounts of detail seen in real life point clouds

Impressive

Okay

Images from Wang,Sakurada, Kawaguchi (2016)

Point Cloud challenges

Varying amounts of detail seen in real life point clouds

Impressive

Okay

Classify what?!

Images from Wang,Sakurada, Kawaguchi (2016)

Required Properties for Ideal 3D Models

Extra invariance criteria needed compared to 2D images.

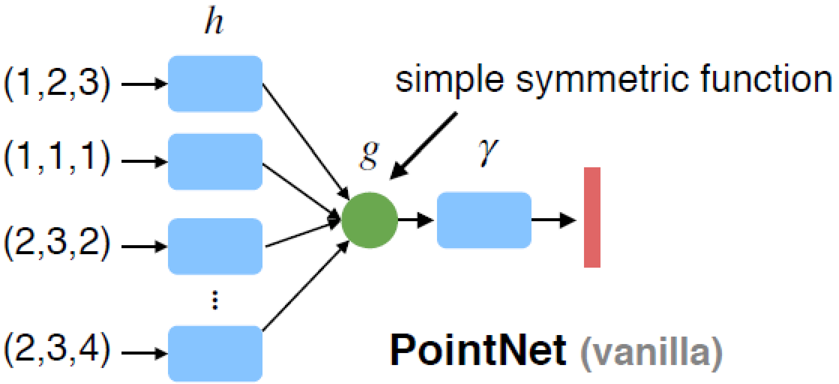
1. **Invariance to permutations (*n*!)**
   - Point set is unordered by nature

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

2. **Invariance under rotations**
   - Different viewing angles of a motorcycle is still a motorcycle.
   - Rare orientations in data also usual in automotive applications (intersections, unexpected traffic situations, special terrain conditions, …)

PointNet[1] Approach for Invariance Properties

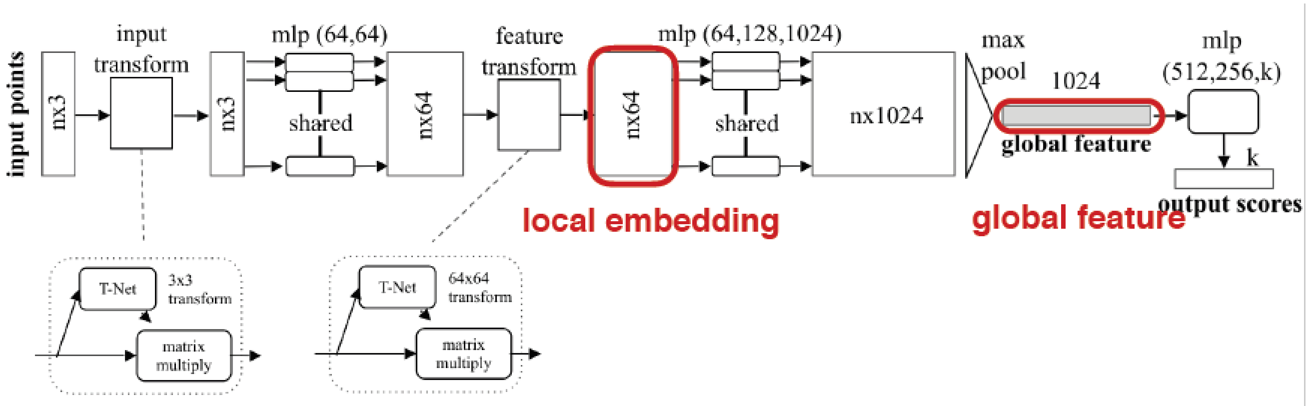
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



- h(): affine transformation (by fully connected)
- g(): symmetry function (by max fn)
- γ(): affine transformation (by fully connected)

[1] Qi, Su, Mo, Guibas: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

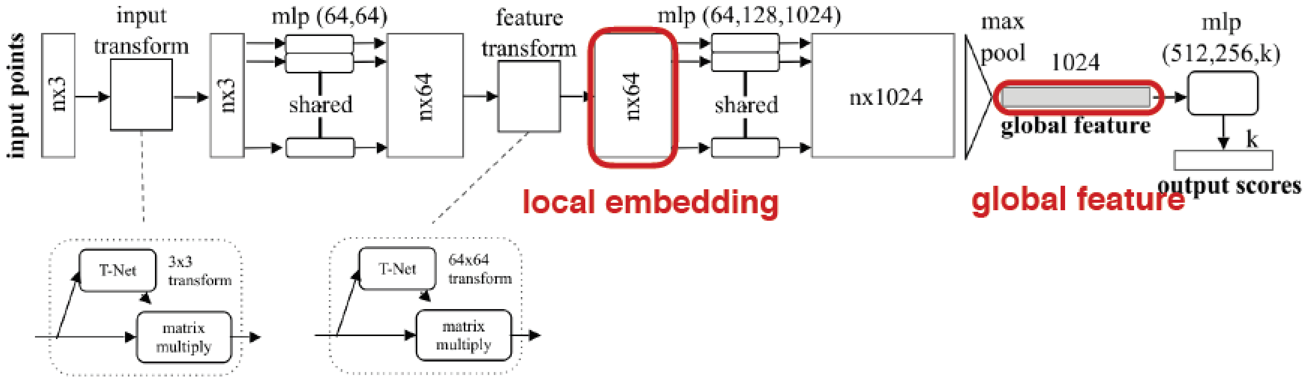
PointNet Extended Architecture



- **T-Net is a mini Point-Net predicting a rotation matrix**
  - Claims to add extra invariance under rotations.

- **Questions**
  - Is PointNet *really* rotation invariant?
  - What does T-Net learn? A canonical orientation or a perspective where it works well?

PointNet Extended Architecture



- T-Net extensions add *a lot* of parameters with minimal accuracy gain!
- Automotive needs:
  - safety-critical
  - embedded (small compute)
  - real time

	#params	FLOPs/sample
PointNet (vanilla)	0.8M	148M
PointNet	3.5M	440M

Transform	accuracy
none	87.1
input (3x3)	87.9
feature (64x64)	86.9
feature (64x64) + reg.	87.4
both	**89.2**

Black-box model assumptions

We need to test all assumptions in safety-critical systems!

Can we just assume we an recognize the same object in different orientations?

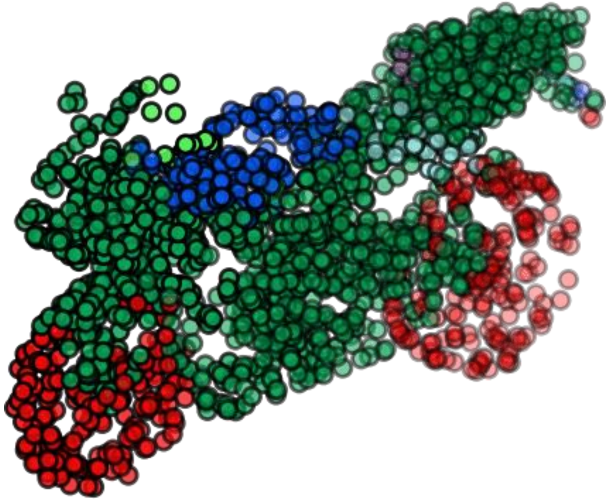


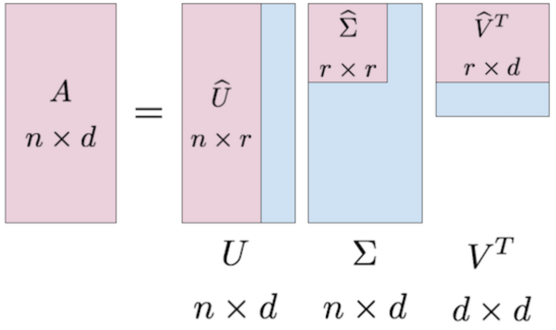
Image from Shen,Feng, Yng, Tian (2017)

Testing the rotation invariance assumption rigorously

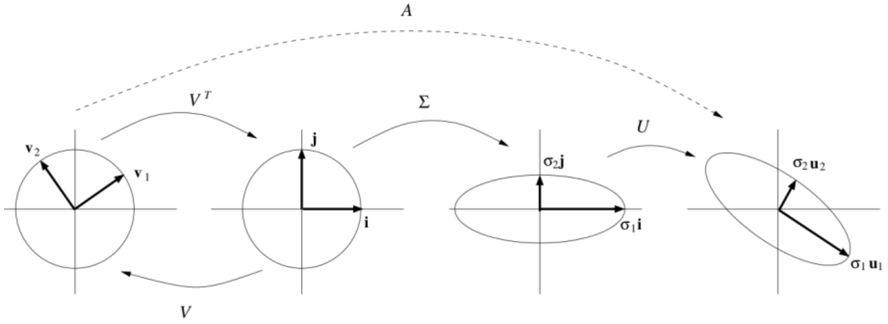
- Create a truly canonical orientation of the data.
  - The tool: Singular Value Decomposition

$$A = U\Sigma V^T$$

$$A_{canonical} = \Sigma V^T$$

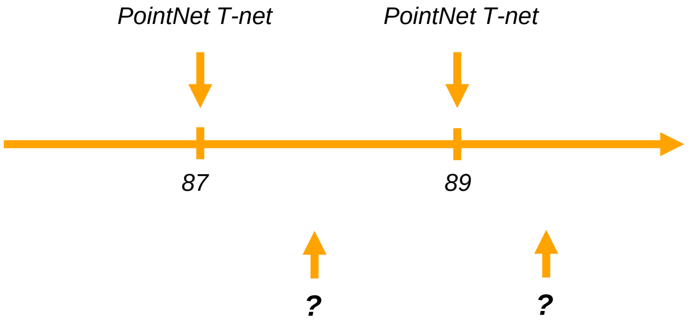


- *Orientation alignment used both training and test/prediction time.*
- *Intuitively for 3D shapes: do an SVD on the tensor containing the input shapes and undo the data's observed orientation leaving the inherent variance-based orientation.*



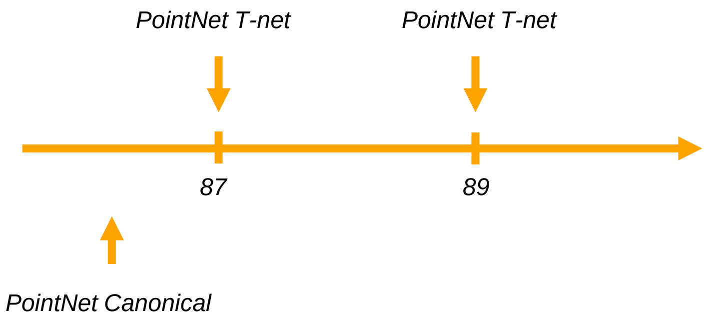
Testing the rotation invariance assumption rigorously

- How would a network with explicit canonical orientation behave compared to the basic ones?



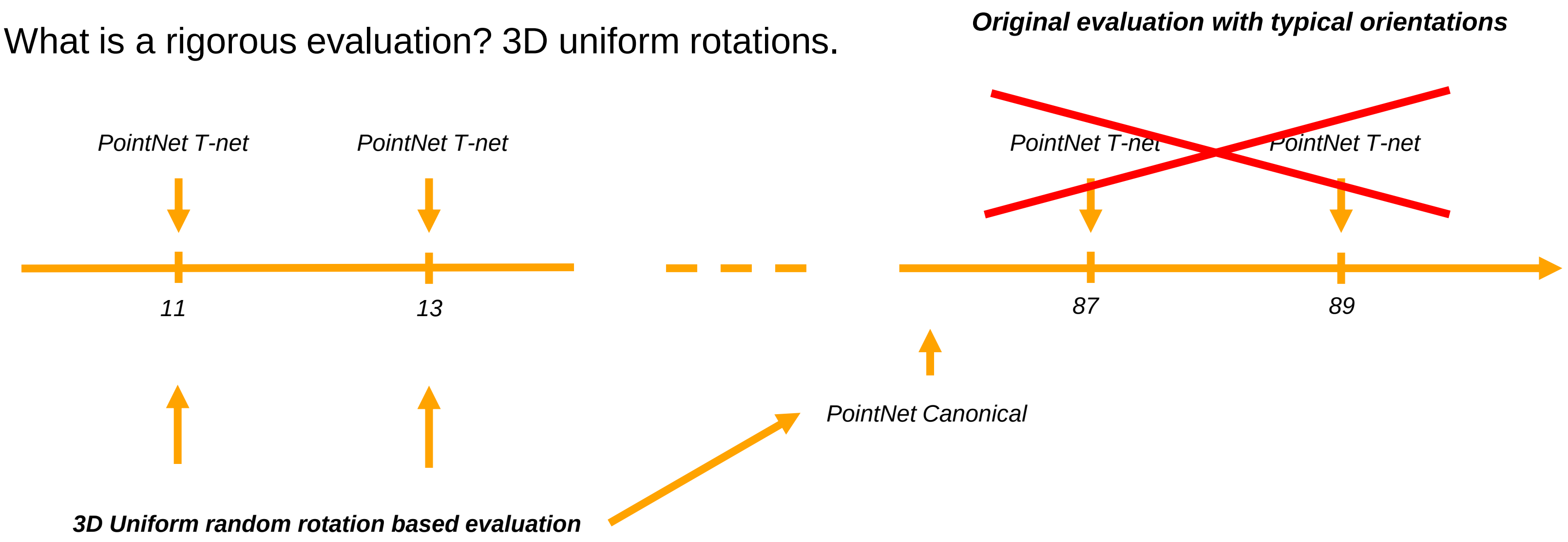
Testing the rotation invariance assumption rigorously

- How would a network with explicit canonical orientation behave compared to the basic ones?



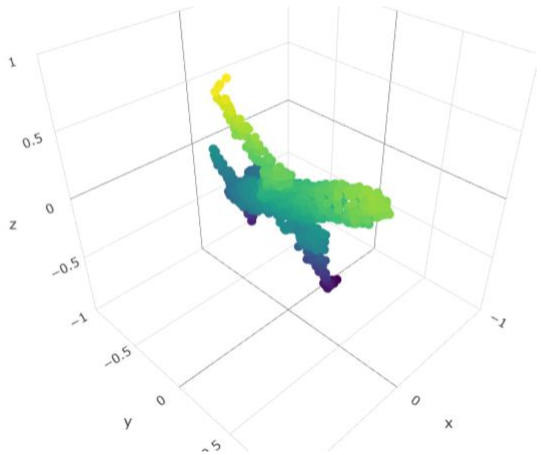
Testing the rotation invariance assumption rigorously

- How would a network with explicit canonical orientation behave compared to the basic ones?
- What is a rigorous evaluation? 3D uniform rotations.



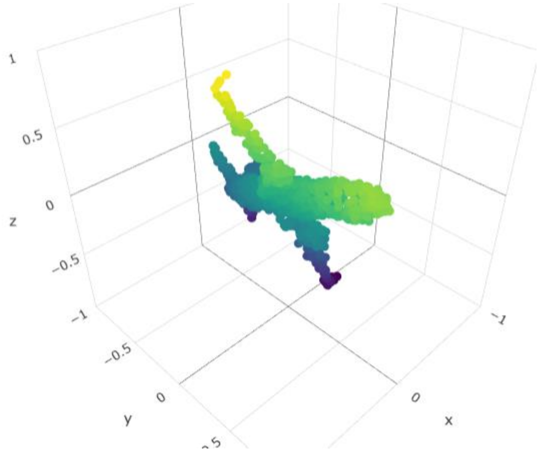
Testing the rotation invariance assumption rigorously

Input

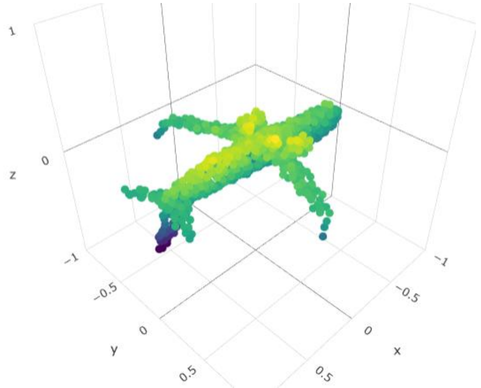


Testing the rotation invariance assumption rigorously

Input

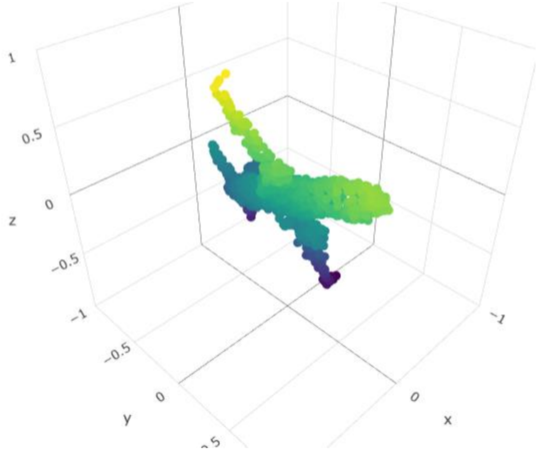


$$A_{canonical} = \Sigma\, V^T$$



Testing the rotation invariance assumption rigorously

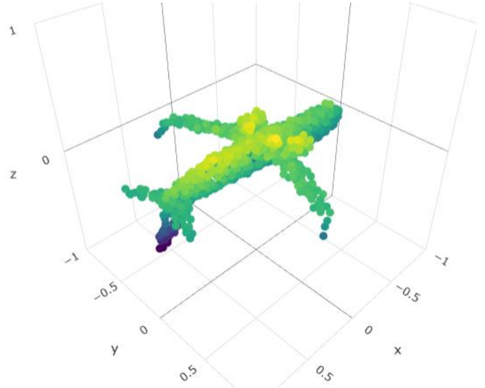
Input



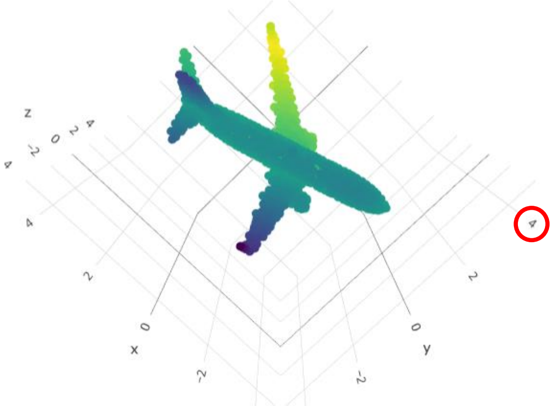
T-Net does not produce canonical orientations!

```
transformation matrix(T):
 [[ 4.000807   -1.3027827   0.12525207]
 [ 1.3016961   2.255568   -3.2670841 ]
 [ 2.5019124   3.281554    1.7621716 ]]
T'*T:
 [[23.96043505  5.93404257  0.6571577 ]
 [ 5.93404257 17.55342642 -1.74964532]
 [ 0.6571577  -1.74964532 13.79477535]]
det(T):  72.26053072374529
3rd root of det(T):  4.165179436892269
```

$A_{canonical} = \Sigma\, V^T$



T-Net



Evaluation Results: Rotation Invariance

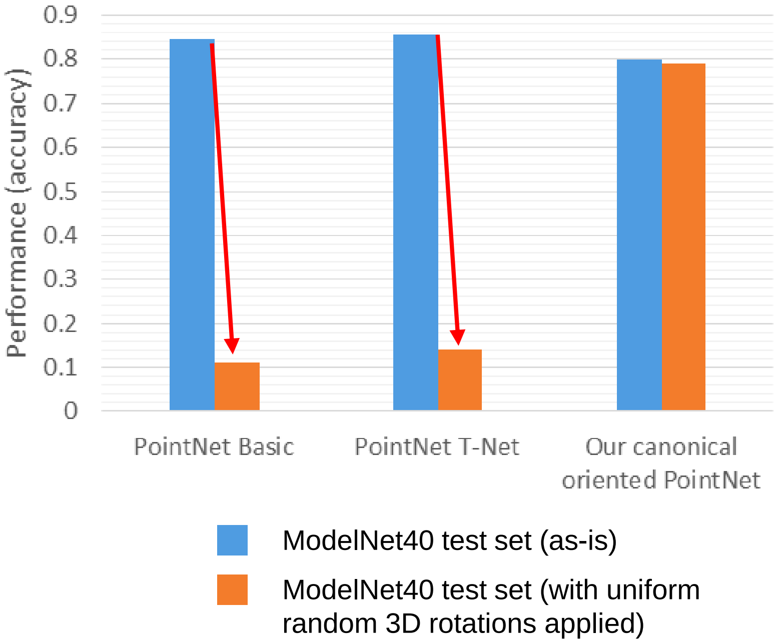
Is PointNet really rotation invariant (paper‘s claim)?

- Based on experiments: no, **far from it**.
- Both Basic and T-Net versions collapse for 3D random viewing angles.

How does our SVD canonical orientation perform?

- Results slightly below PointNet on *unrotated* test data.
- Virtually no degradation in accuracy when test data is rotated randomly – strongly outperforms PointNet baselines.

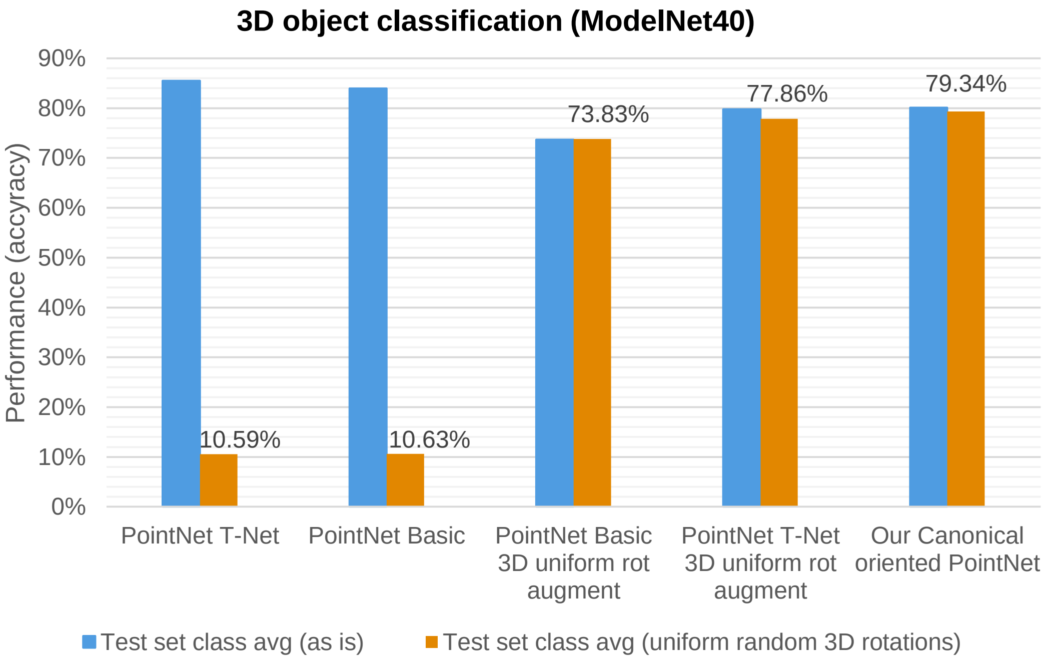
3D object classification (ModelNet40)



Adding data augmentation for Rotation Invariance

Can we handle cases where test time canonical rotation is not an option?

- Possible with well designed data augmentation
- For this use case we used 3D uniform rotated data augmentation for training
- Slight decrease in accuracy compared to our Canonical oriented one
- **Important use case:** well designed data augmentation is an option for scenes



Conclusions

Our study shows

- One of the most popular approaches is *not rotation invariant* for 3D.
  - Model fails with more general orientations.
  - We need to be careful with model design in 3D, e.g. fallen pedestrians, bikes, motorcycles.
- PointNet‘s T-Net addon still does not fix this, but increases the #params by > 4x
- We need to dissect models and re-design evaluations to be rigorous and explicitly test all assumptions.
- Our proposals offer **robust 3D rotation invariance** outperforming basic PointNet variants by a large margin.

Safe and Dynamic Driving towards Vision Zero

http://www.continental-jobs.com

Sense**PlanAct**