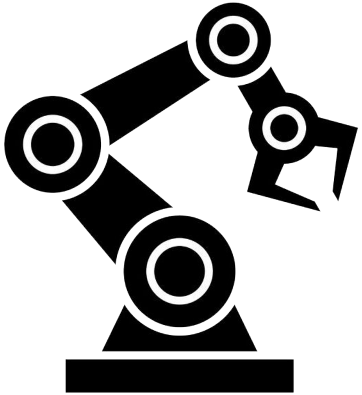
Unsupervised Classification of Sensor Data

A Learning Based Approach

Team: **The Boring Panda**

Members: **Daniel & Giorgio**



Our goal

Classify each data reading as self, background or other using Unsupervised Learning techniques.

Hypothesis: the perception task can be solved without relying explicitly on geometry and configuration.

Summary

- We sampled random position and orientation for the end effector
- We framed the perception problem as a time series problem
- We built a statistical model
- We built a library
- We built an online visualization tool

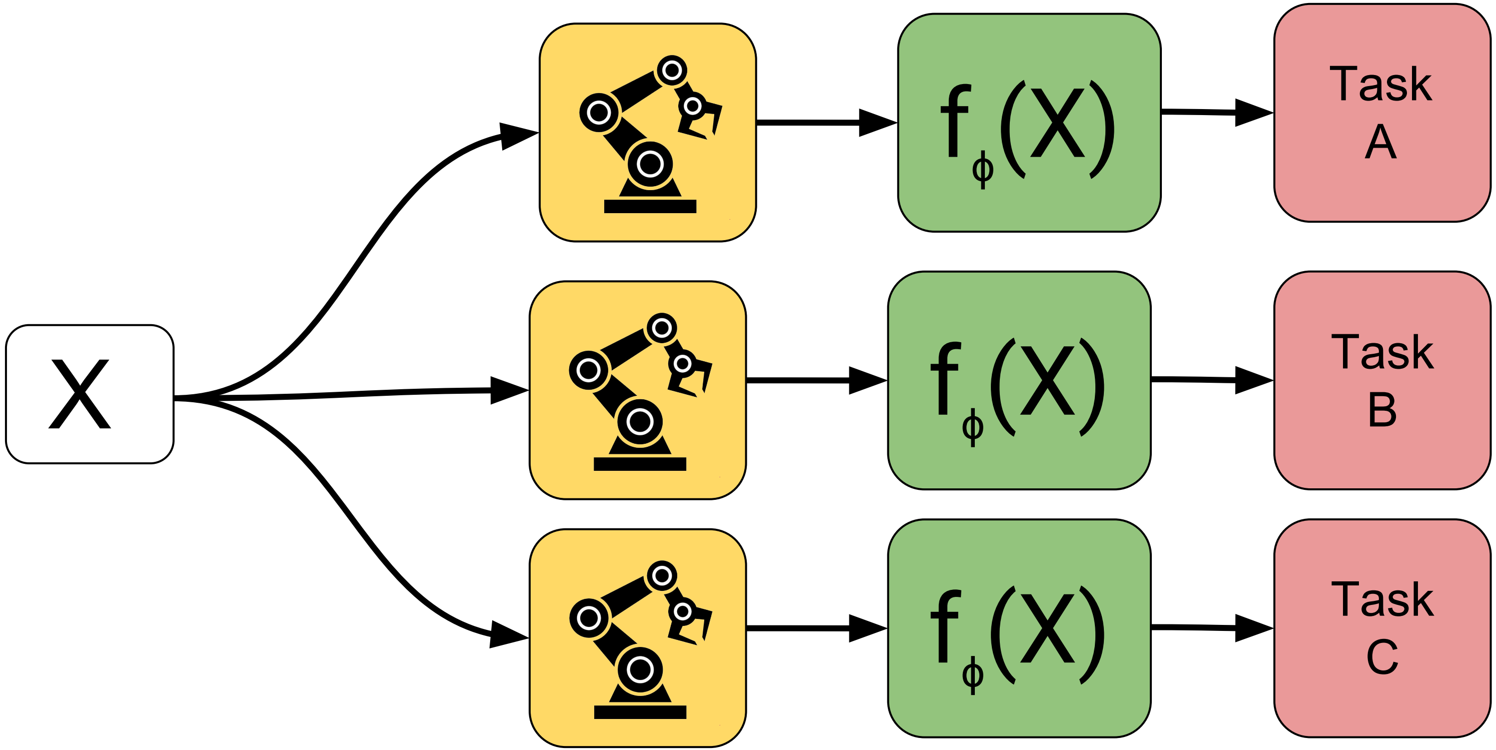
Learning based vs Geometry based

Geometry based: we build a specific model of our system, and on top of it we use machine learning to solve specific tasks.

Learning based: we build a statistical model able to deal with a dynamic environment, and on top of it we inject geometrical constraints and task dependent informations.

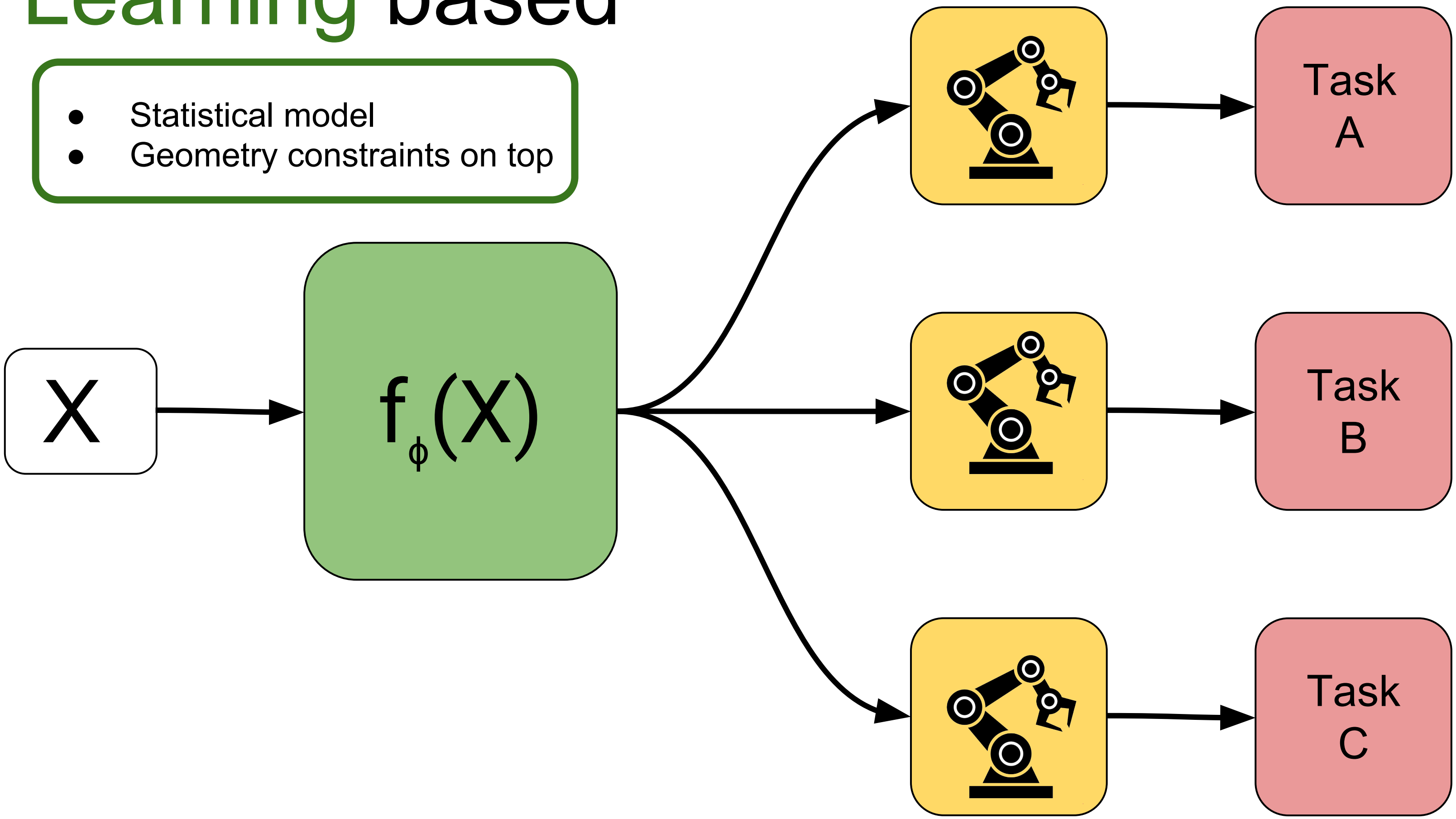
Geometry based

- Geometrical model of our system
- Machine learning on top



Learning based

- Statistical model
- Geometry constraints on top



Why Unsupervised?

Labelling sensor data is:

- Challenging
- Time consuming

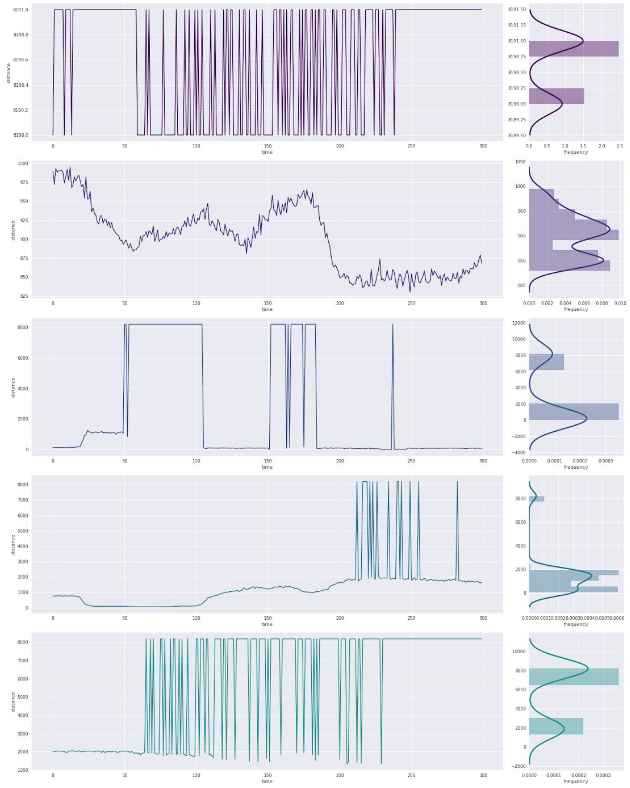
If we use heuristics to label the data, then we learn these heuristics.

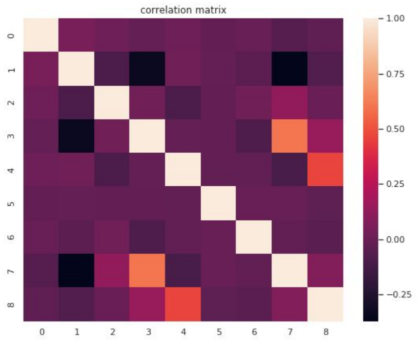
Statistical Modeling

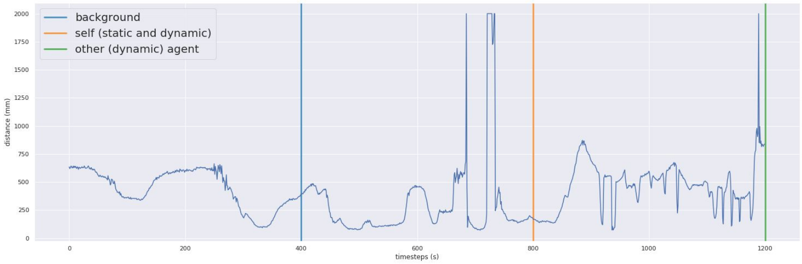
With a probabilistic approach:

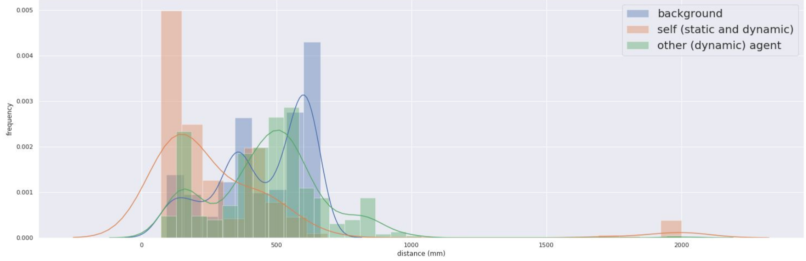
- We can model the uncertainty
- We can quantify the uncertainty of our predictions
- We can deal with anomalies in the environment

Multi-modality



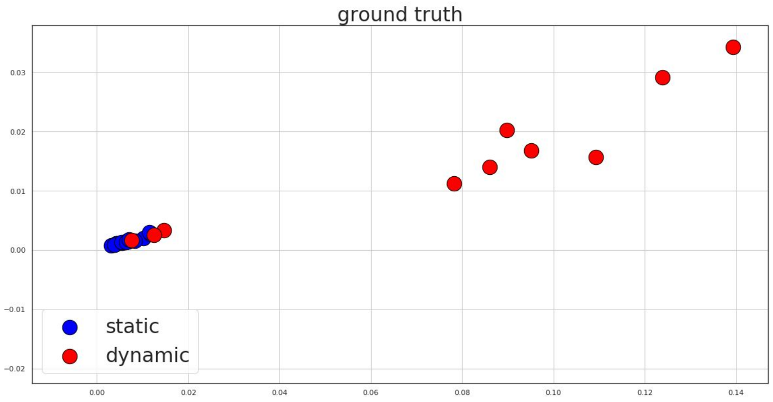




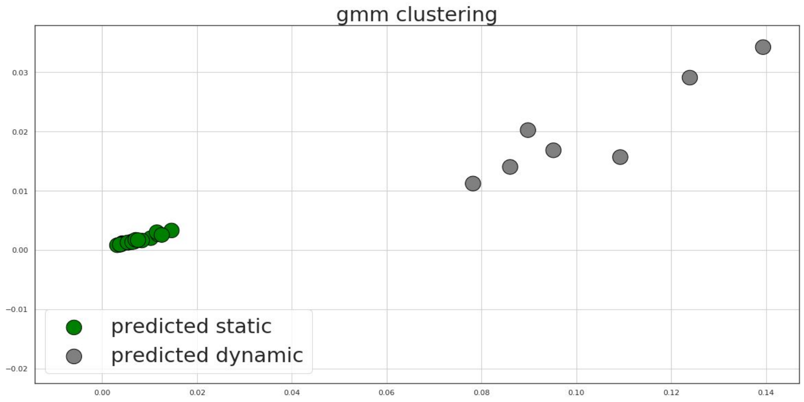


GMM clustering with handcrafted feature

Ground truth



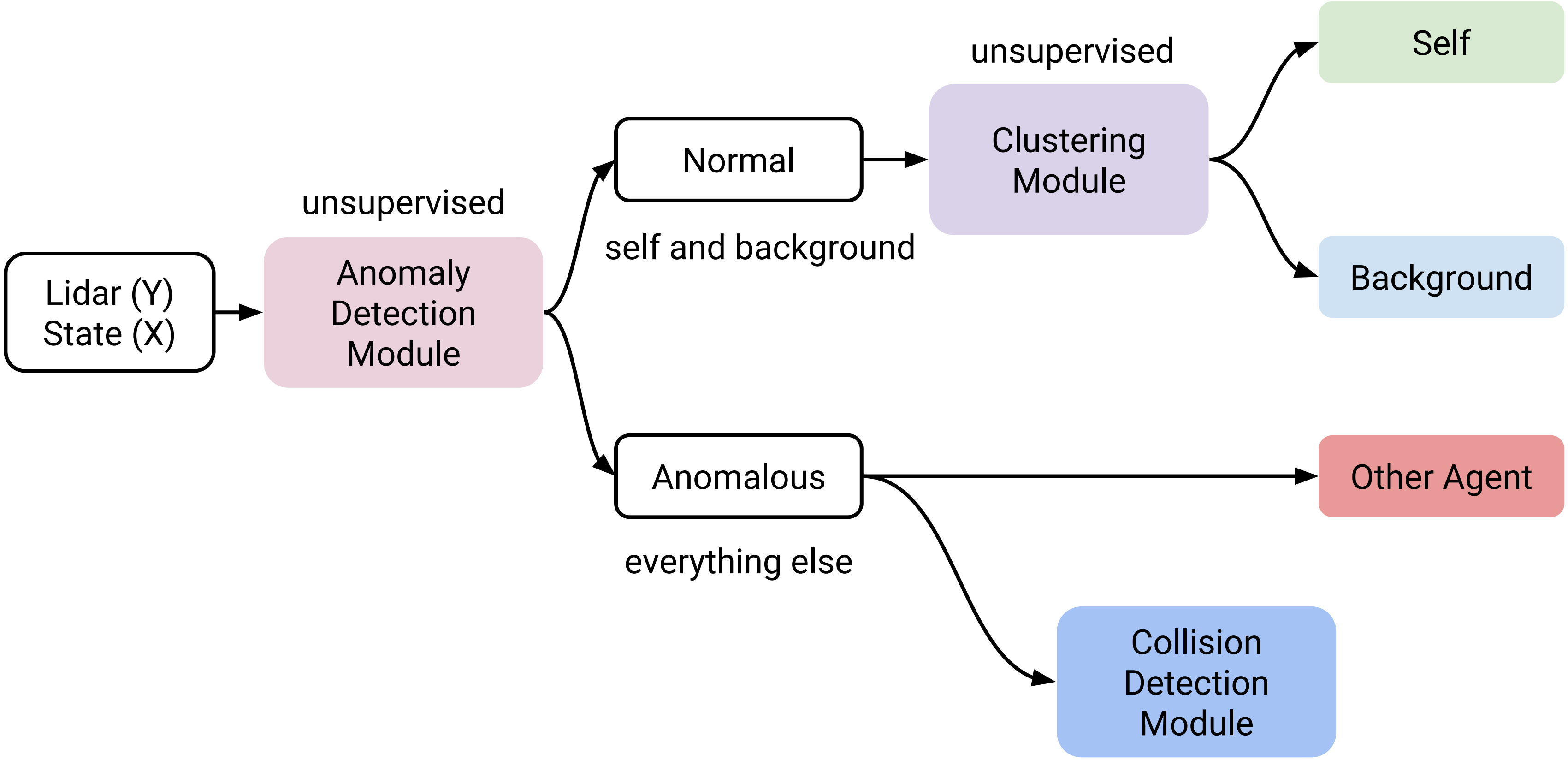
Prediction



Input $x_{1:T}$ is a time series →create a new time series $x'_{1:T}$ by abs(x(t) - x(t+1)) →feature vector $\phi(x'_{1:T})$

where $\phi_1 = \max(x'_{1:T})$, $\phi_2 = \mathrm{std}(x'_{1:T})$

Sensing Framework



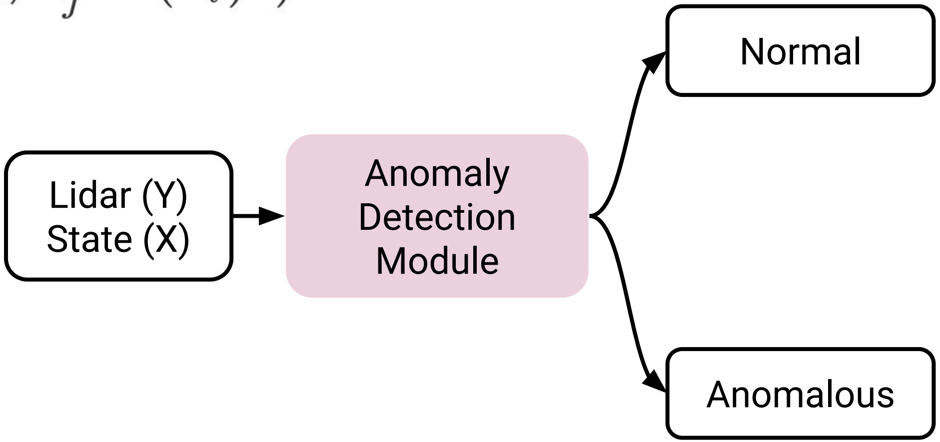
Anomaly Detection Module

$$p(Y|X) = \prod_i p(y_i|x_i)$$

$$= \prod_i \prod_j \mathcal{N}(y_{ij}^{lidar}|\mu_j^{lidar}(x_i), \sigma_j^{lidar}(x_i)^2)$$

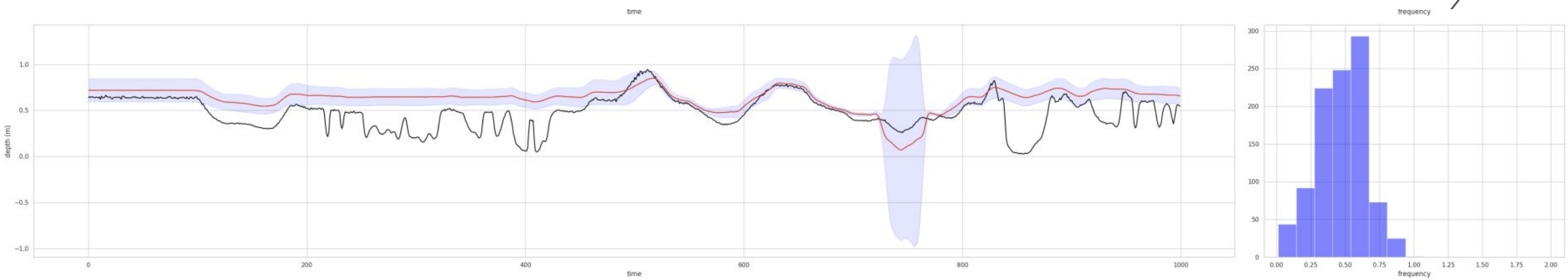
For each lidar one separate mean and variance

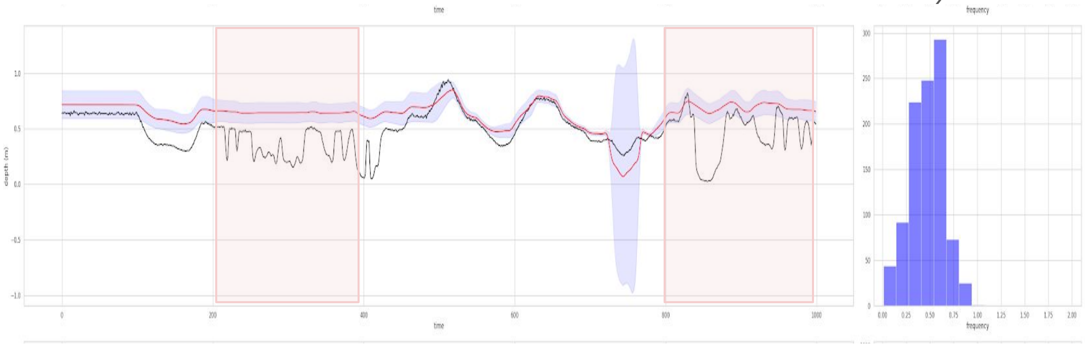
$$L(Y, X) = -\sum_i \log p(y_i|x_i)$$



Anomaly Detection Module

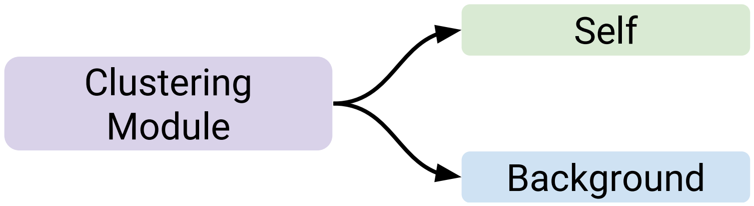
While moving robot arm, Person was moving hand in front of lidar 3 in various directions





Clustering Module

Main assumption: the signals can be **clustered** with a **mixture**.



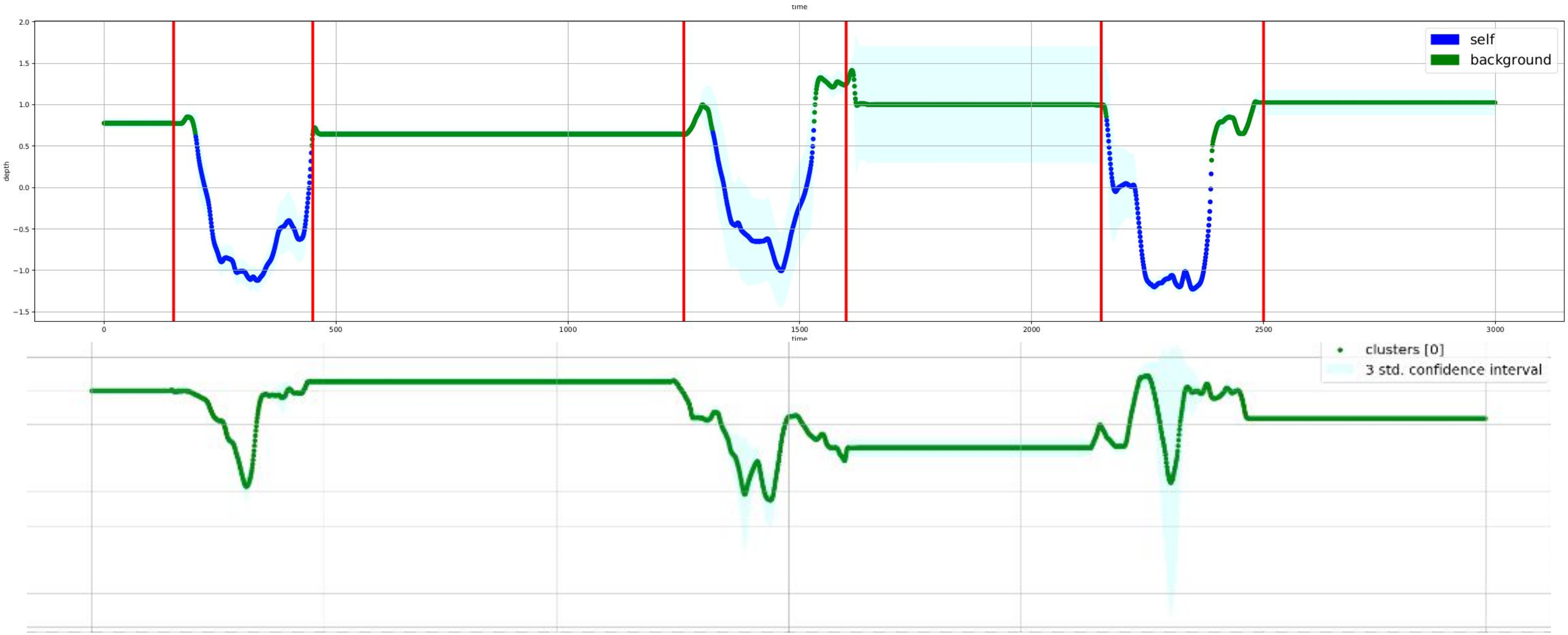
$$p(Y|X) = \prod_i p(y_i|x_i)$$

$$= \prod_i p(y_i^{lidar}|x_i)$$

$$= \prod_i \prod_j \mathcal{N}(y_{ij}^{lidar} | \sum_k \pi_{jk}^{lidar}(x_i)\mu_{jk}^{lidar}(x_i), \sum_k \pi_{jk}^{lidar}(x_i)\sigma_{jk}^{lidar}(x_i)^2)$$

Clustering Module

Lidar 3 known gt



- Global Accuracy: 0.89
- Self (TP): 0.69 (for worst possible gt)
- Background | self (FN): 0.30

Clustering on a controlled experiment. Lidar n.3 reading. The red vertical lines represent the limit of ground truth for self.

Collision Detection Module

- (X, Y, C) train set
- Perturb 5 % lidar readings randomly around 50 mm (collision threshold)
- Assign class 0 to all unperturbed points
- Assign class 1 to all perturbed points
- Train a discriminator with strong supervision
- Evaluate

$$L(Y, X, C) = -\sum_{i}\sum_{k} c_{ik} \log \pi_k(y_i, x_i)$$

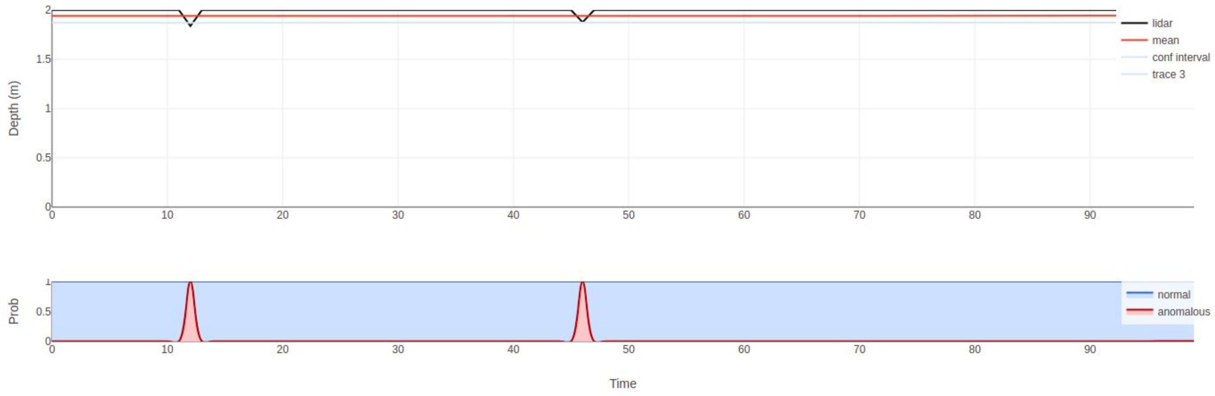
Collision Detection Module

	0	1	2	3	4	5	6	7	8	Avg
N	100	180	140	180	100	170	200	160	180	156.6
Recall	0.96	0.33	0.22	0.56	0.84	0.97	0.22	0.20	0.57	0.54
IoU	0.92	0.32	0.20	0.50	0.73	0.91	0.19	0.17	0.50	0.49
F1	0.96	0.49	0.33	0.67	0.84	0.95	0.32	0.29	0.66	0.61

Result on 2000 point with ground truth for collision detection

Visualization Tool

- [normal]
- [anomaly]
- [clustering]



Code, experiments, visualizations, demo: [repo]

Conclusions

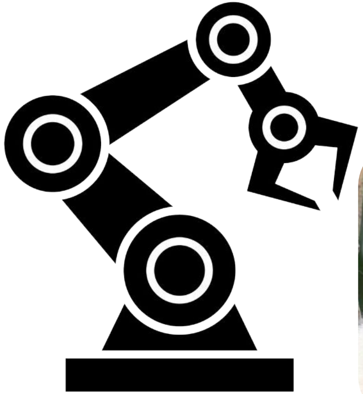
Advantages of a Statistical approach

- We can reuse our ML models
- We can quantify the uncertainty of our predictions

Drawbacks

- Not easy to interact with the environment
- Not suited for precision tasks
- We didn’t solve the perception task in a general sense

Thanks!



Unsupervised
Classification
of
Sensor Data

A Learning Based Approach

Powered by Daniel & Giorgio