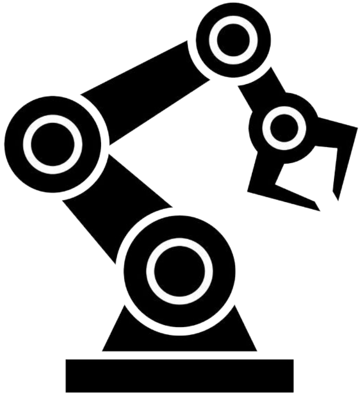
Unsupervised Perception of Dynamic Environments

A Learning Based Approach

Team: **Boring Panda**

Members: **Daniel, Giorgio**



Our goal is to sense the environment.
Our approach is:

- Learning Based
- Explicit Geometry free
- Explicit Configuration free
- Partial knowledge of the Environment

Why Unsupervised?

Labelling sensor data is:

- Challenging and time consuming
- If we use heuristic, we learn heuristic, no structure
- Classical supervised machine learning assumes static environment between train and test set (no domain shift)

Why no Geometry and Configuration?

1. Do we think Geometry is not relevant?
2. Do we think we can interact with the environment without Configuration?
3. Do we think that (Statistical) Machine Learning will solve any problem?

Why no Geometry and Configuration?

- Domain Knowledge
- Technical Challenge
- Application and task specific
- Fragile in Dynamic Environments

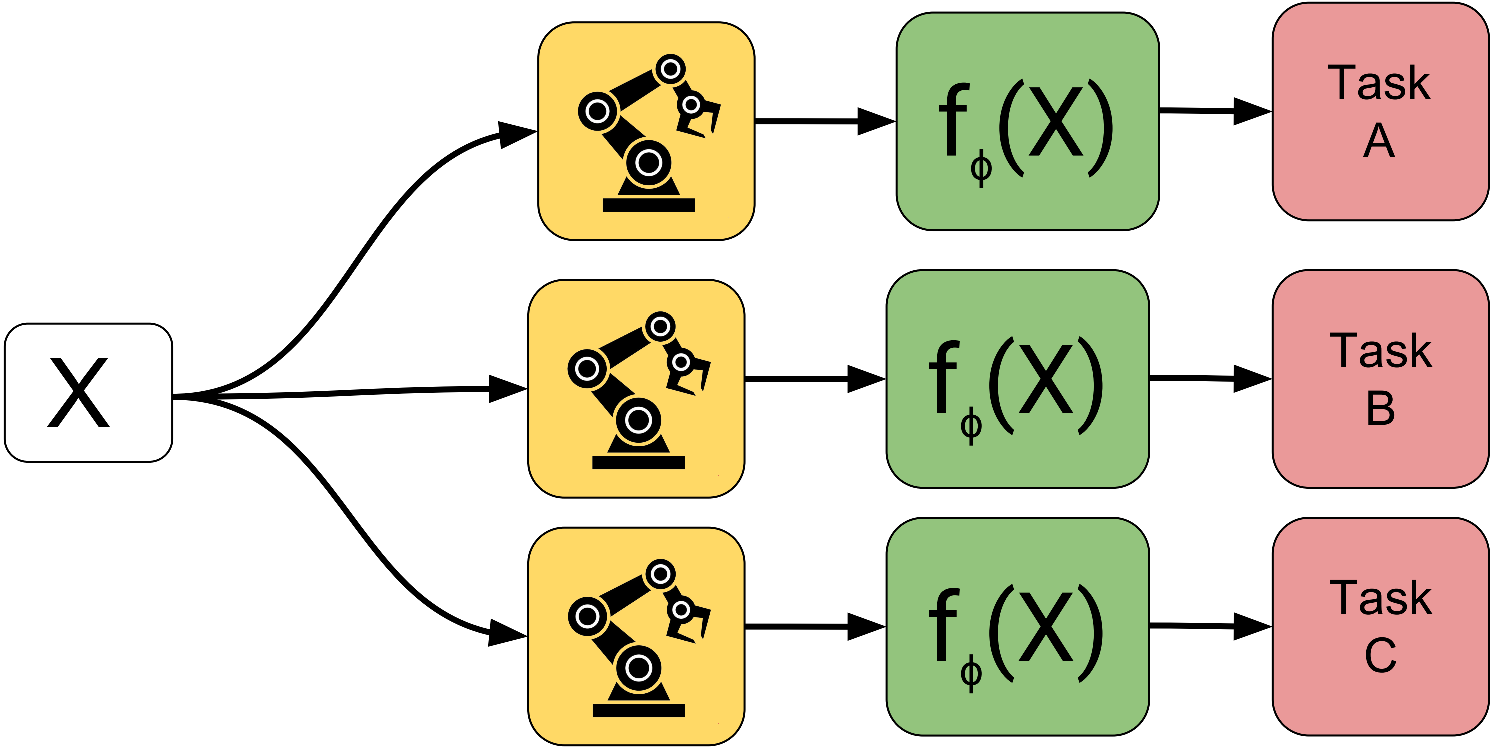
Learning based vs Geometry based

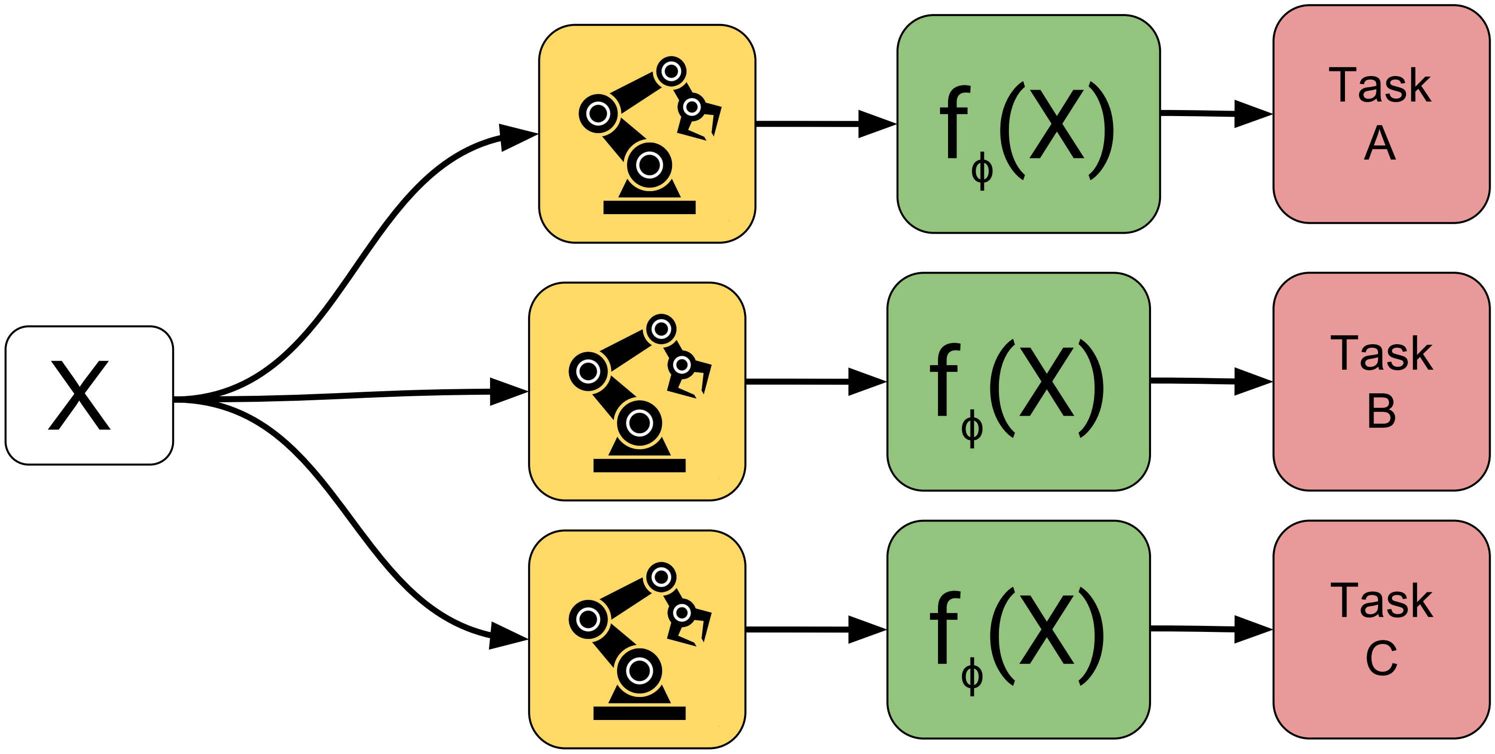
Geometry based: we build a specific model of our system, and on top of it we use machine learning to solve specific tasks.

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

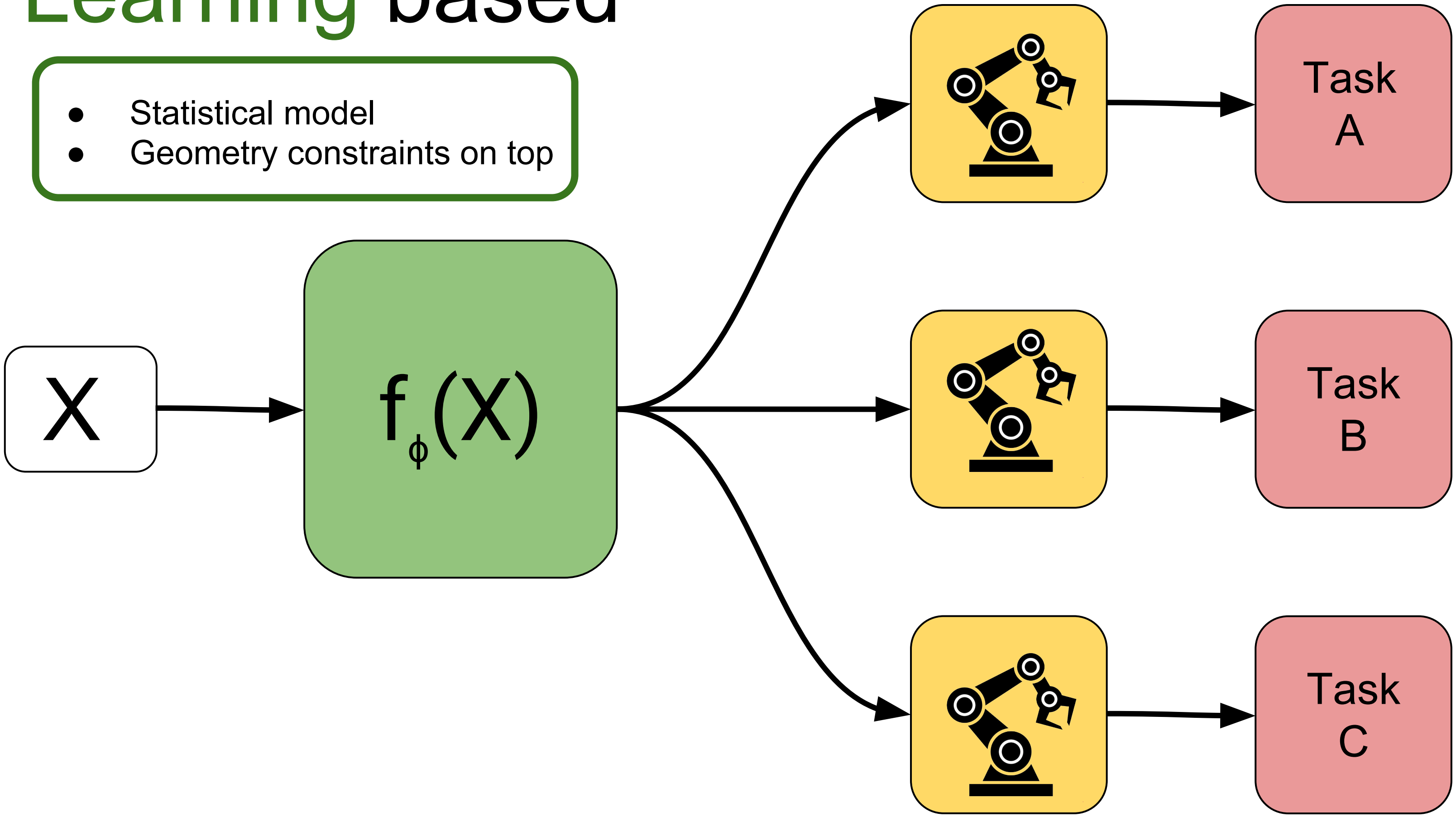
- Geometrical model of our system
- Machine learning on top

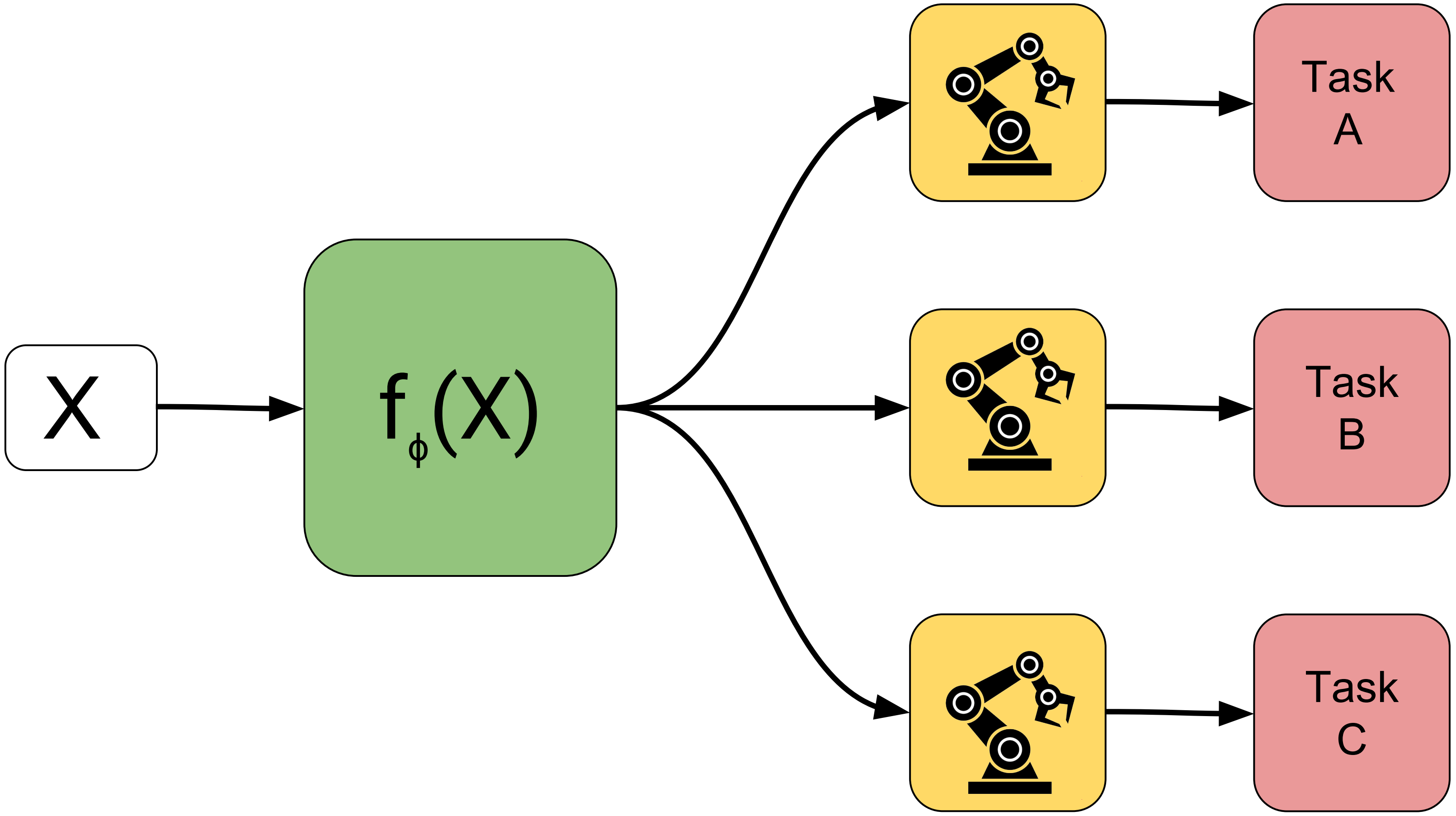



X
$f_\phi(X)$
Task A
$f_\phi(X)$
Task B
$f_\phi(X)$
Task C

Learning based

- Statistical model
- Geometry constraints on top




X
fφ(X)
Task A
Task B
Task C

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
- Need adaptive modeling
- Not suited for specific precision tasks

Our Idea

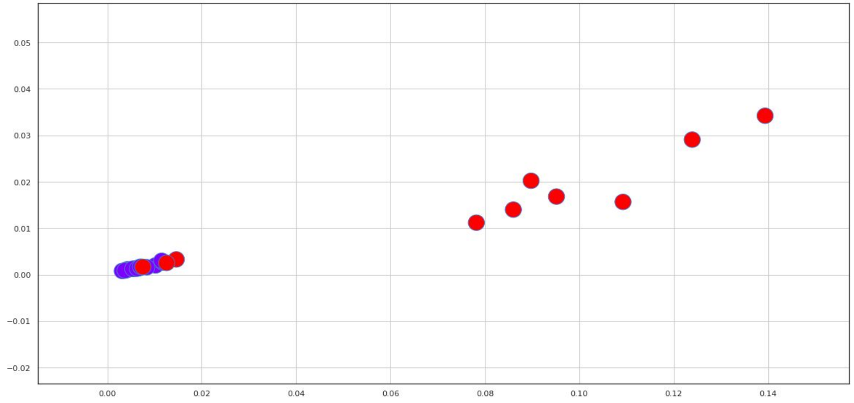
We build a sensing framework, learning based, able to generalize with minimum modifications to different geometries, configurations, data acquisition pipelines.
We want to achieve this result in an unsupervised way, in a Dynamic Environment, quantifying the uncertainty of our models.

In practice

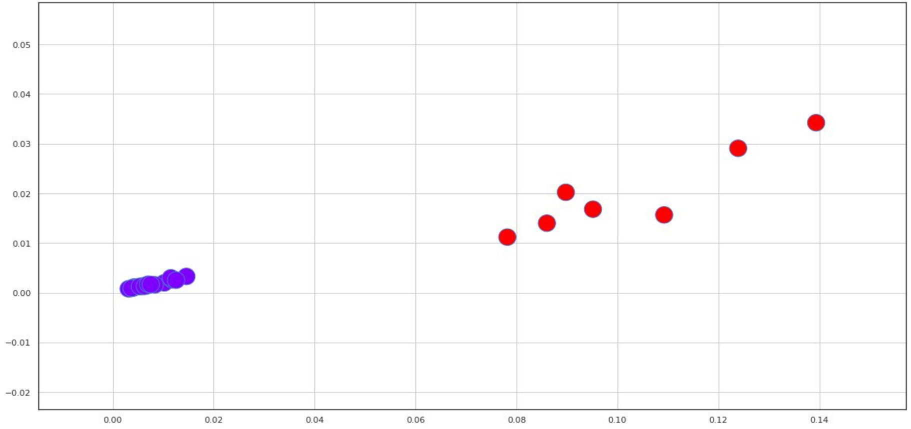
- We sampled randomly points (position and orientation for the end effector) in the environments.
- We framed the perception problem as a time series analysis problem.
- We built a statistical framework.
- We experimented with different models and pipelines.
- We built an online tool to investigate our data.

GMM clustering with handcrafted feature

Ground truth



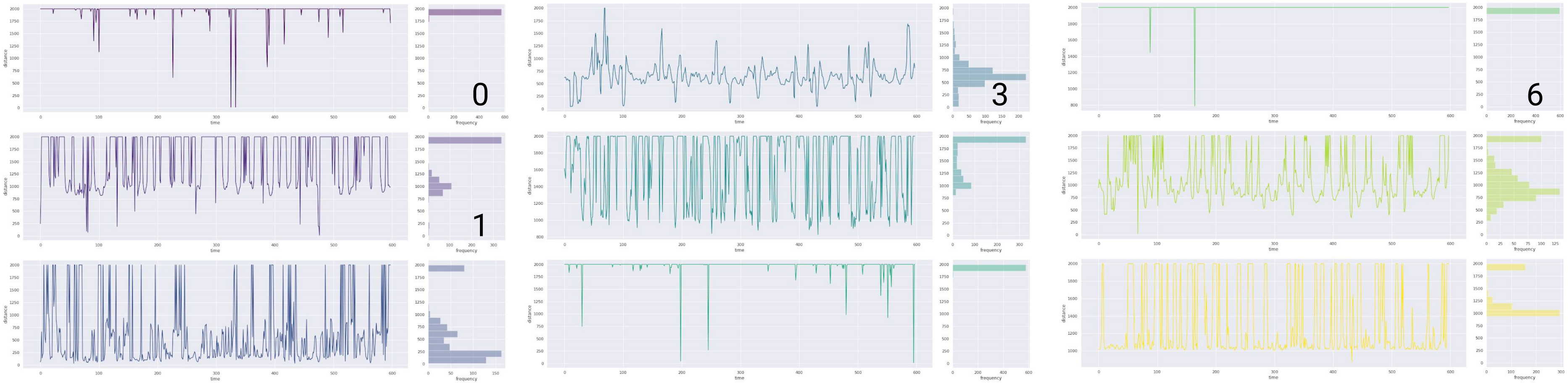
Prediction

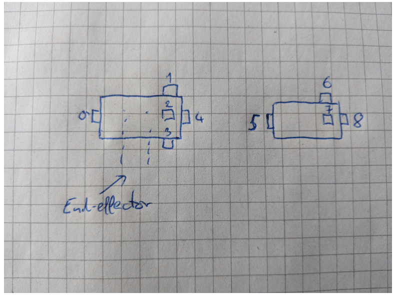


Input $x_{1:T}$ is a time series →create a new time series $x'_{1:T}$ by abs(x(t) - x(t+1)) →feature vector $\phi(x'_{1:T})$

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

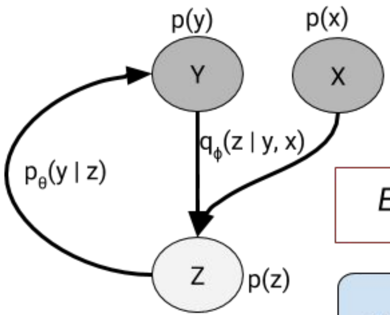
5 minutes lidar measurements of static environment and robot itself





lidar mapping (front view)

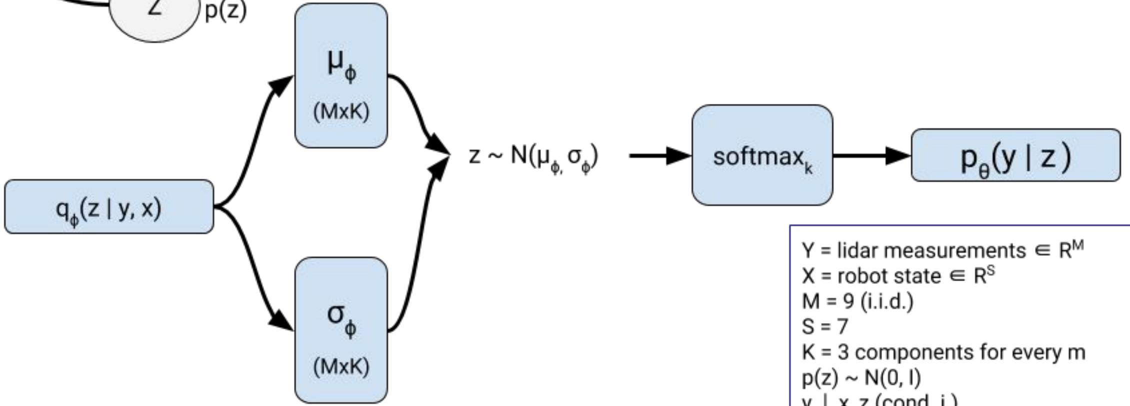
Generative model for unsupervised clustering

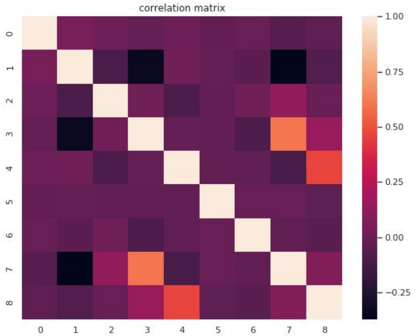


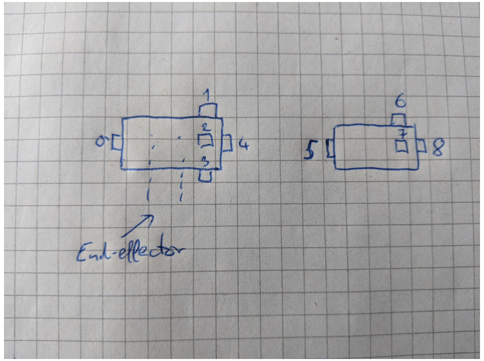
$$p_\theta(y_m \mid z_m) = \sum^K_k N(y_m \mid \mu^k_m, \sigma^k_m)\ s(z_m)$$

$$s(z_m) = \text{softmax}(z_m)_k$$

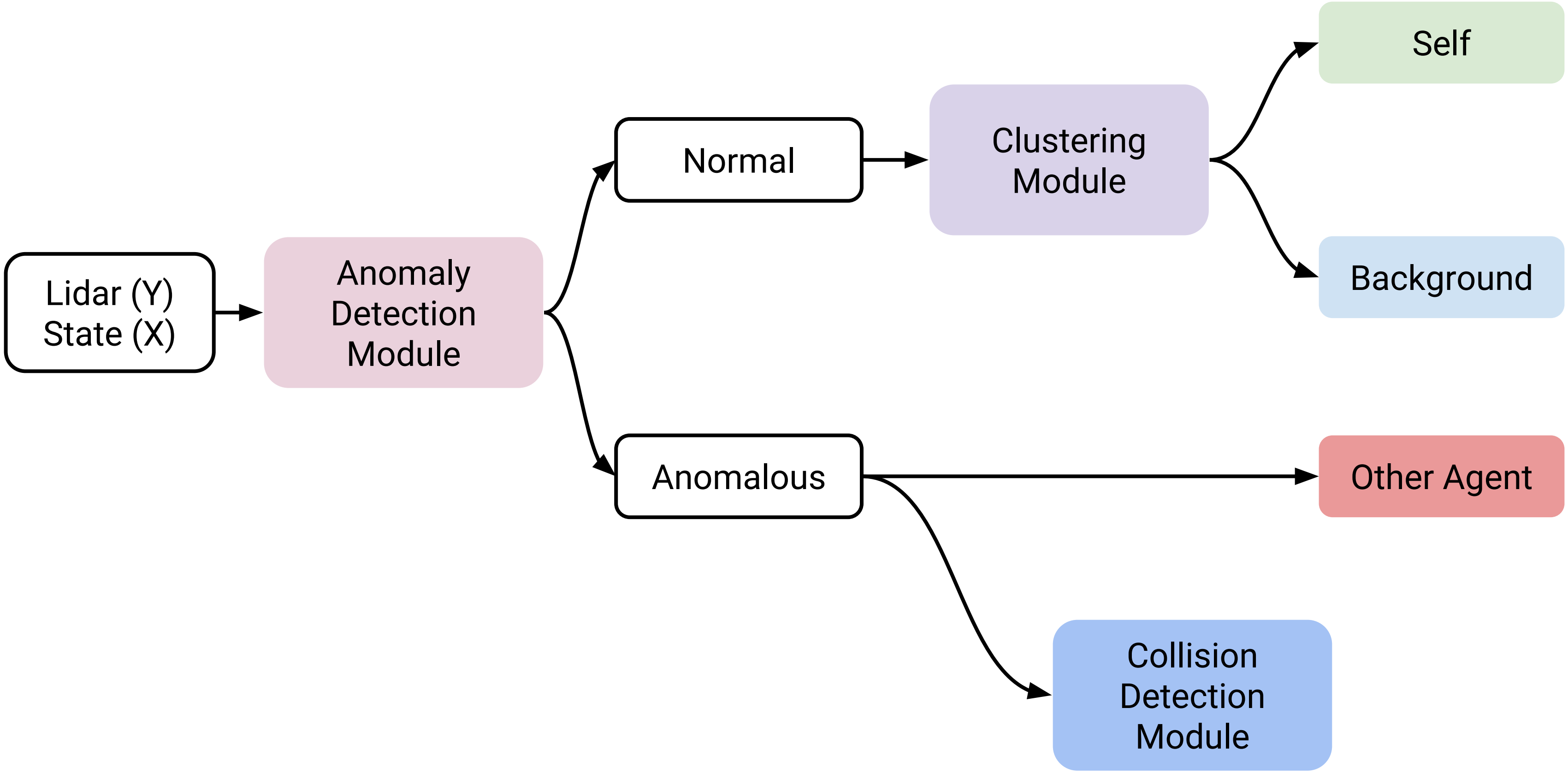
$$\boxed{ELBO(y, x, \phi, \theta) = -KL(q_\phi(z \mid y, x) \,\|\, p(z)) + 1/L \sum_z log(p_\theta(y \mid z))}$$



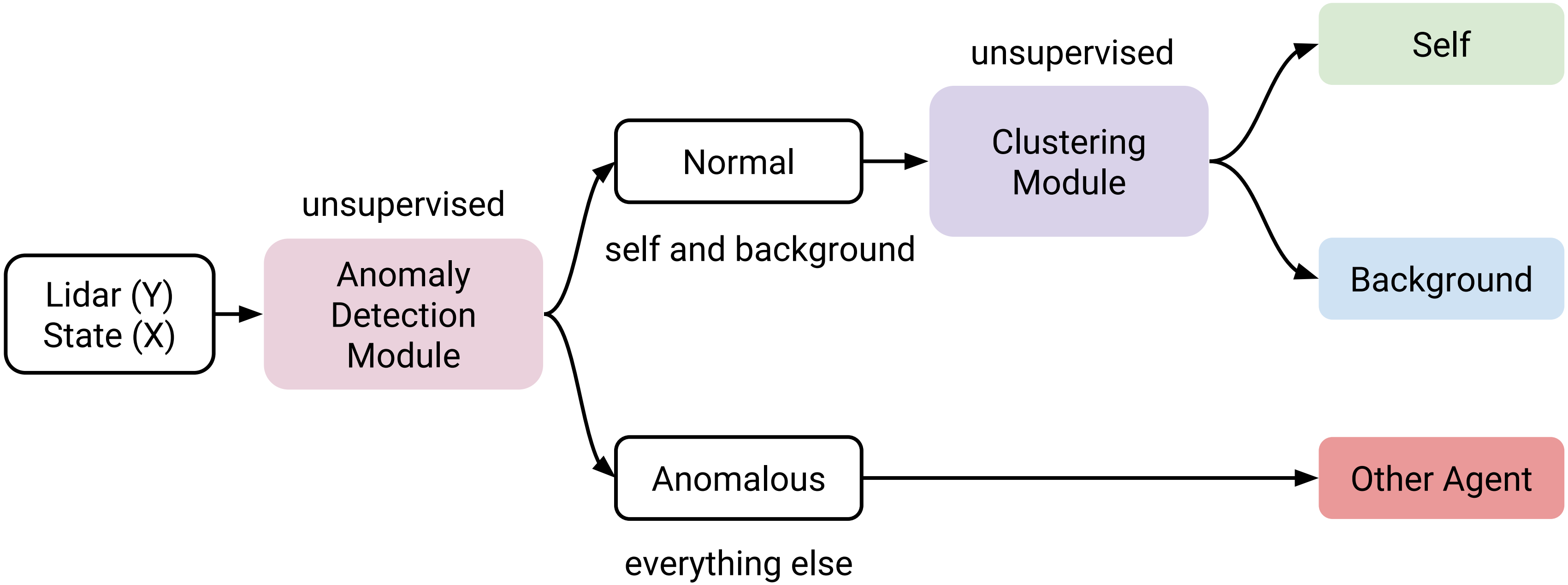




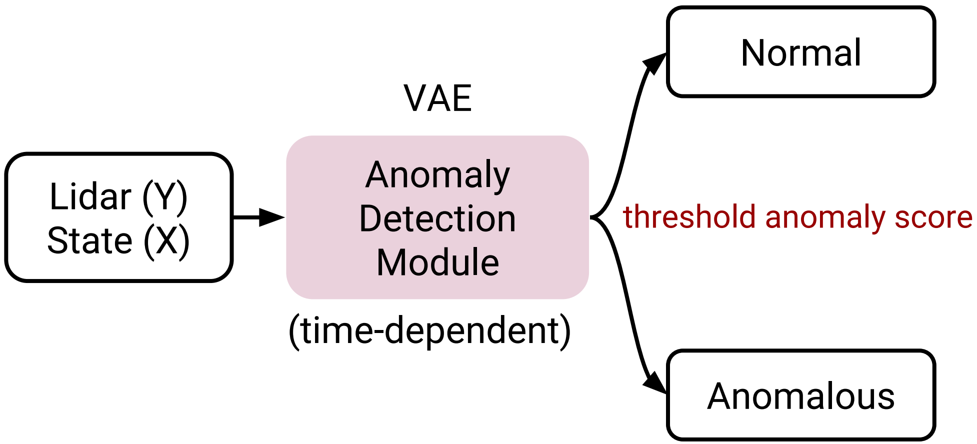
Y = lidar measurements $\in R^M$
X = robot state $\in R^S$
M = 9 (i.i.d.)
S = 7
K = 3 components for every m
$p(z) \sim N(0, I)$
$y \perp x, z$ (cond. i.)


Lidar (Y)
State (X)
Anomaly Detection Module
Normal
Anomalous
Clustering Module
Self
Background
Other Agent
Collision Detection Module

Sensing Framework



Anomaly Detection Module



Notes

- Uncertainty of anomaly score by using the histogram of recon. error on test set
- Threshold anomaly score for final decision
- Diff. btw. Input and recon. May be helpful saying where the anomaly happened
- Plot test time series

Clustering Module

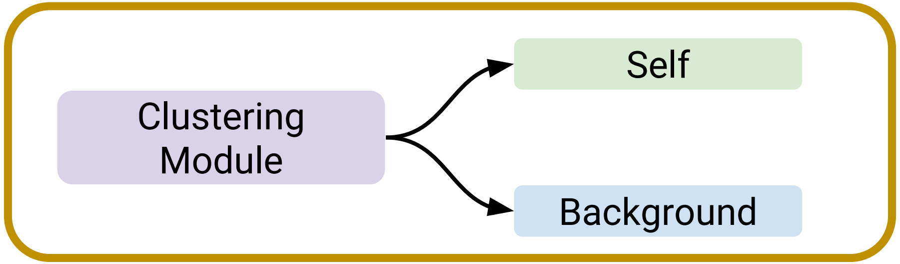
Main assumption: the signals can be **clustered** with some **mixture** in some embedding space.

For the moment:

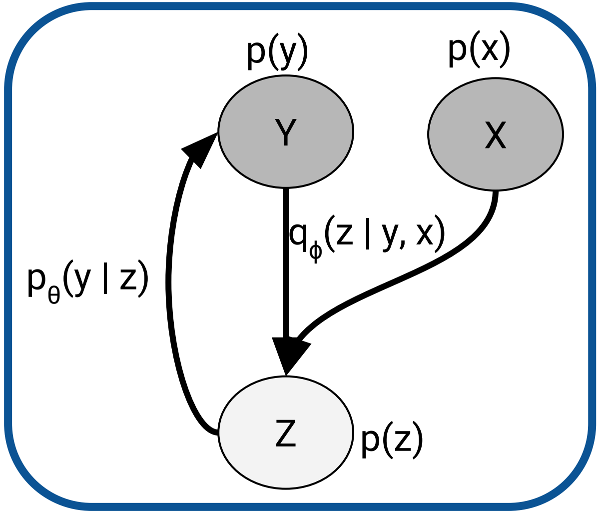
- Regression problem, **regressing** the lidar conditioned on the state.
- Moments weighted sum. The weights are the **deterministic** cluster selectors.

Goal:

- Use a conditional variational autoencoder to cluster **probabilistically** these time series.
- Find an approximation of **q(z | y, x)**



$$
\begin{aligned}
p(Y|X) &= \prod_i p(y_i|x) \\
&= \prod_i N(y_i | \sum_k \pi_{ik}(x_i)\mu_{ik}(x), \sum_k \pi_{ik}(x)\sigma_{ik}(x)^2)
\end{aligned}
$$



AD - Gaussian Regression Model

For each lidar one separate mean and variance

Gaussian conditioned on state

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

Y = lidar measurements
X = joint position

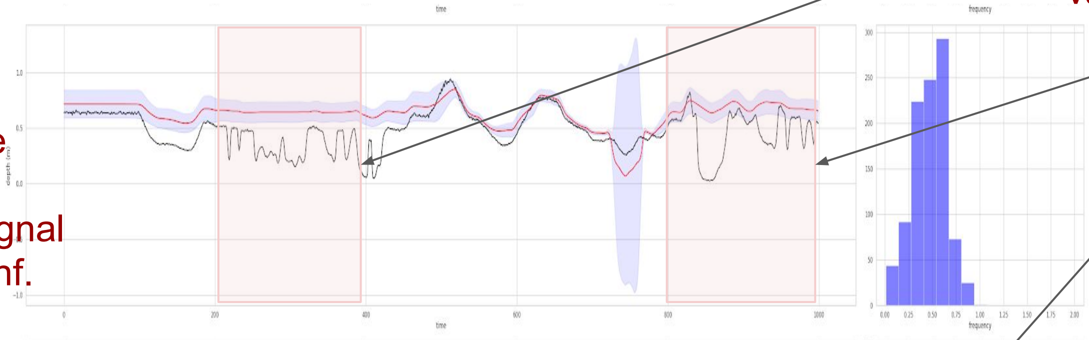
loss

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

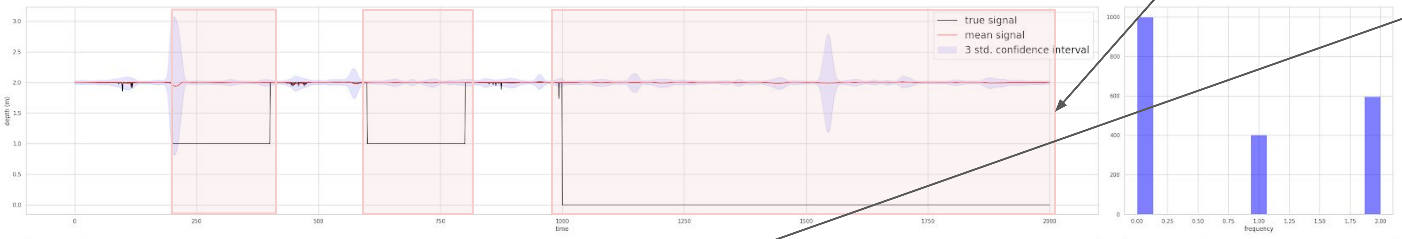
AD - Gaussian Regression Model

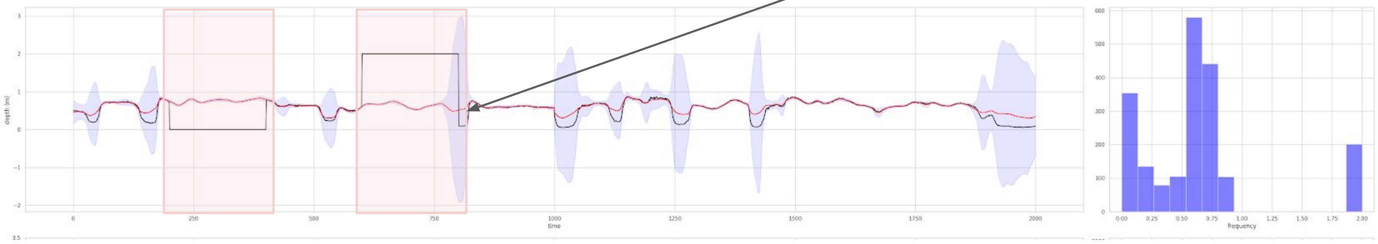
While moving robot arm, Person was moving hand in front of lidar 3 in various directions

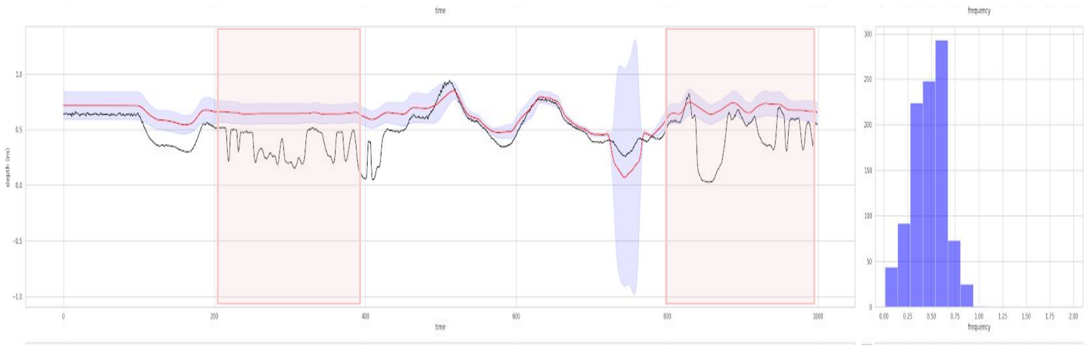
In all cases it can be seen that during an anomaly, the true signal is out of the 3std conf. interval



Synthetic anomaly in test sample by setting signal in specific time interval to certain value

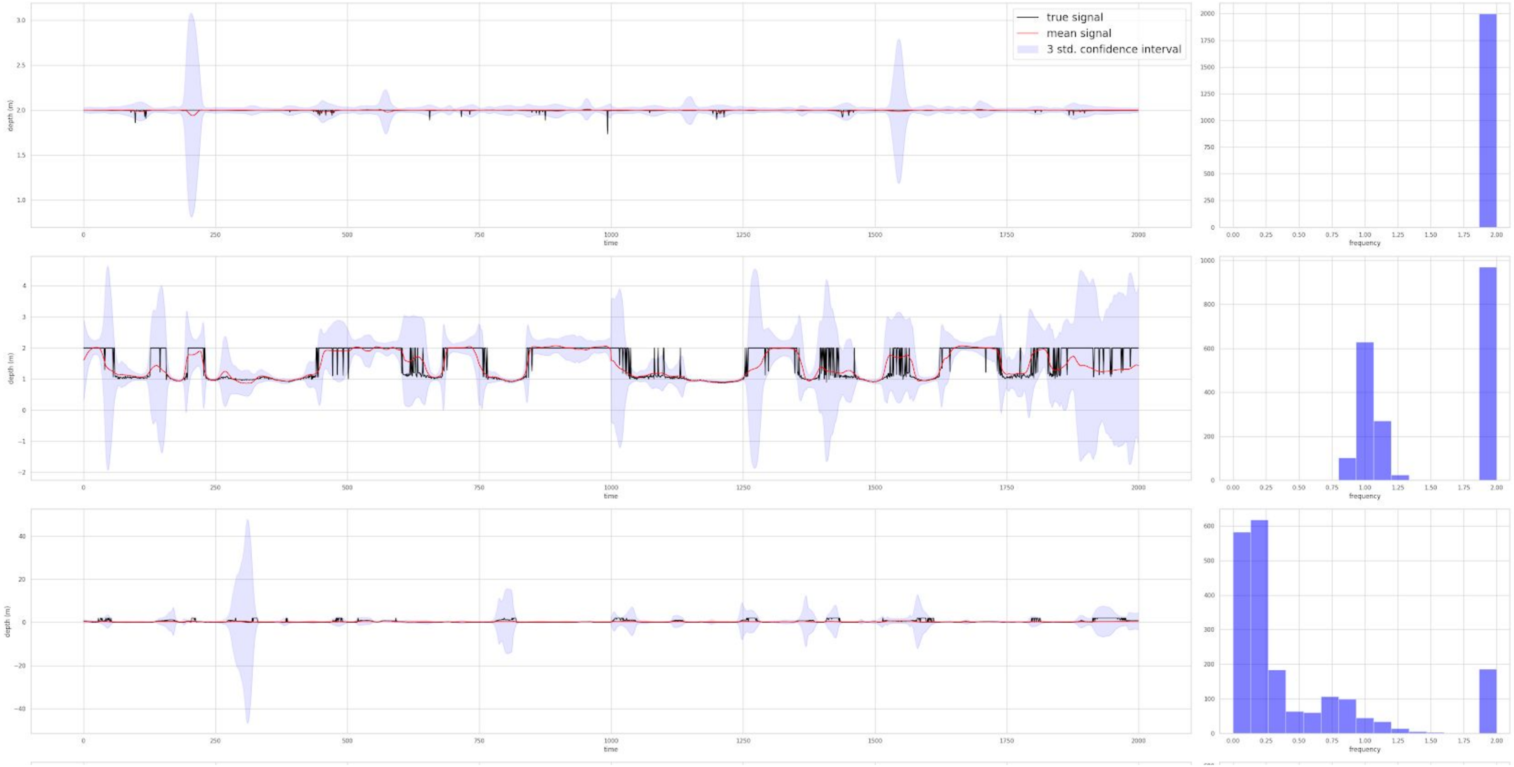


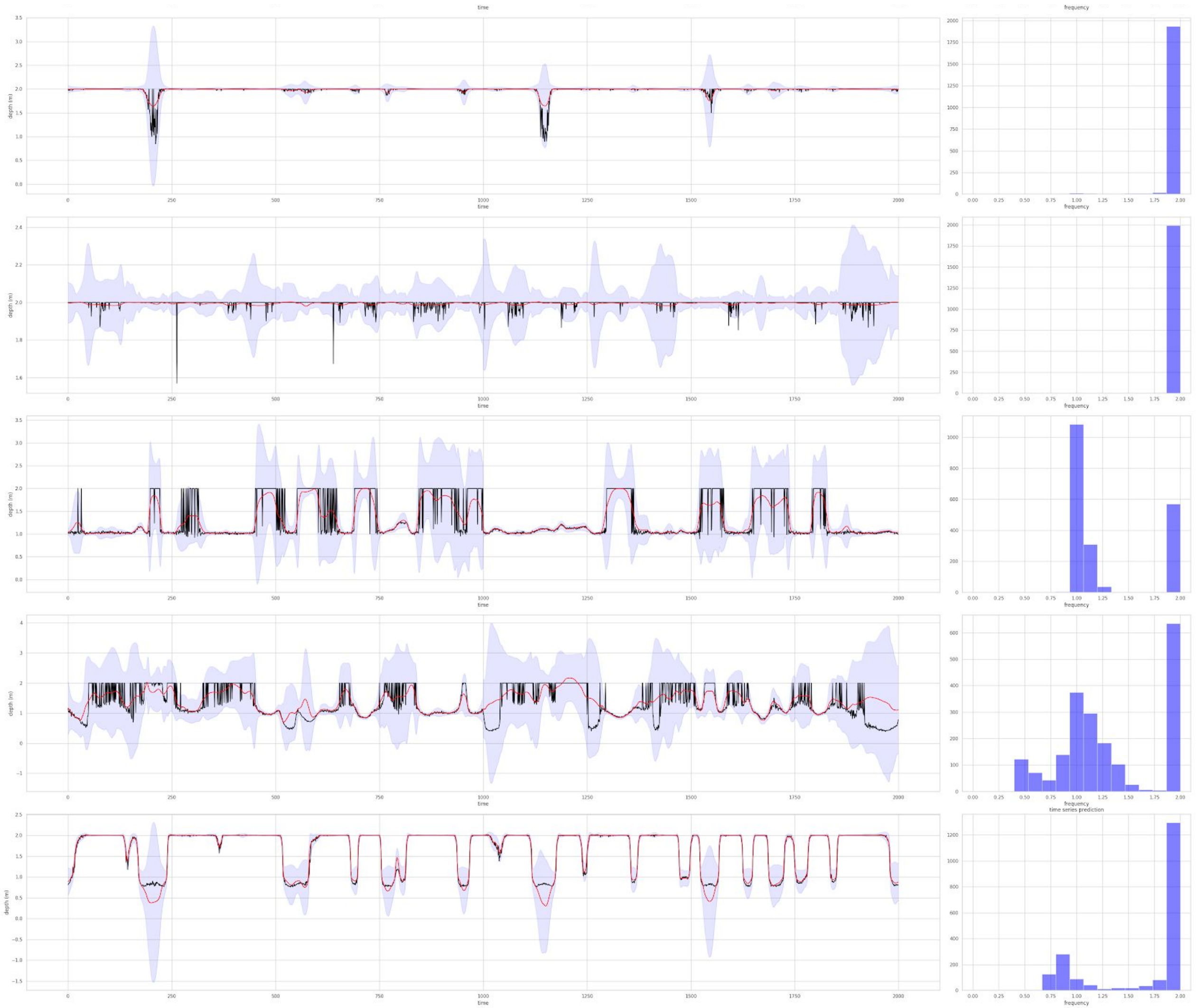



time
frequency
depth (m)
1.0
0.5
0.0
-0.5
-1.0
0
200
400
600
800
1000
300
250
200
150
100
50
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

AD - Gaussian Regression Model

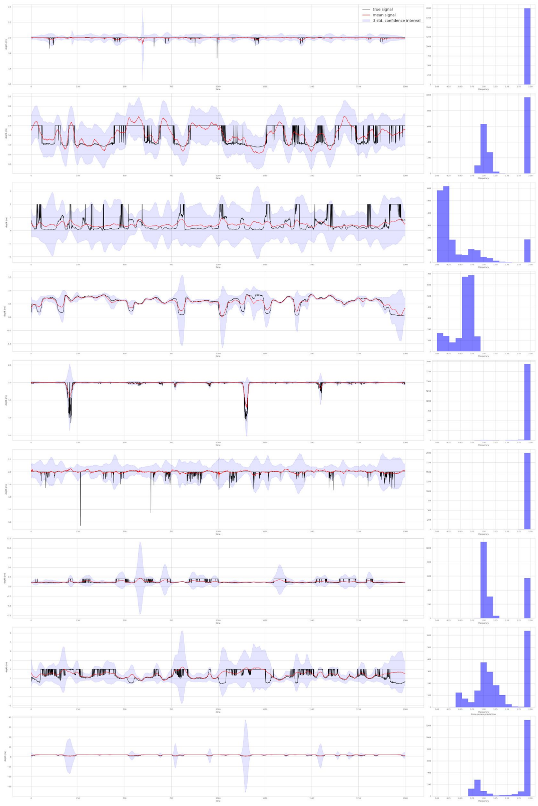
Prediction (red) on test set. Black = true signal, blue = 3std conf. int.



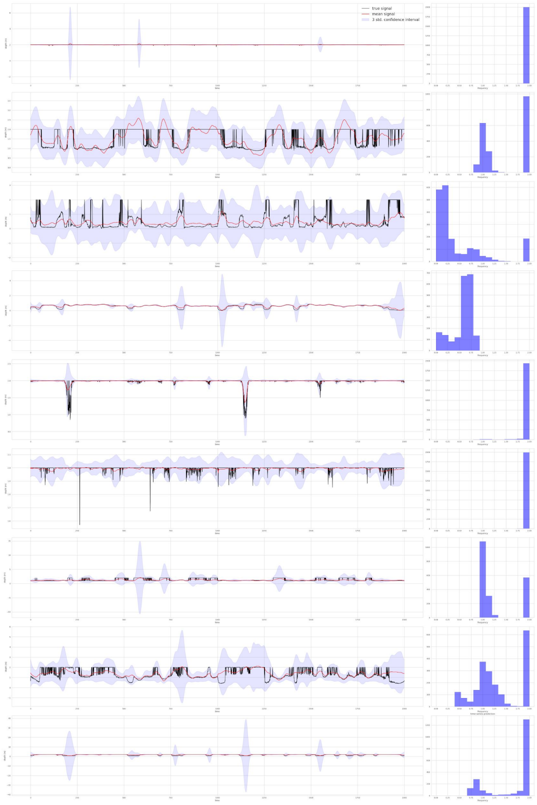


Prediction (red) on test set. Black = true signal, blue = 3std conf. int.

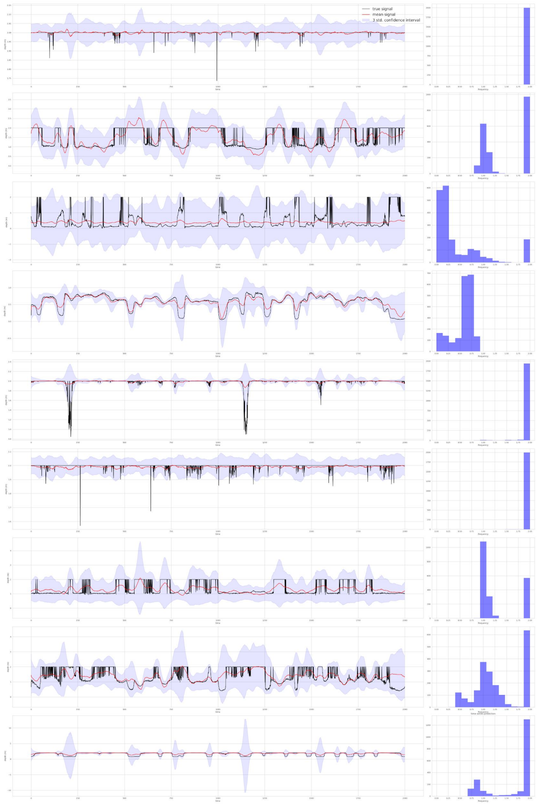
Experiments Model Selector



10 sample x v

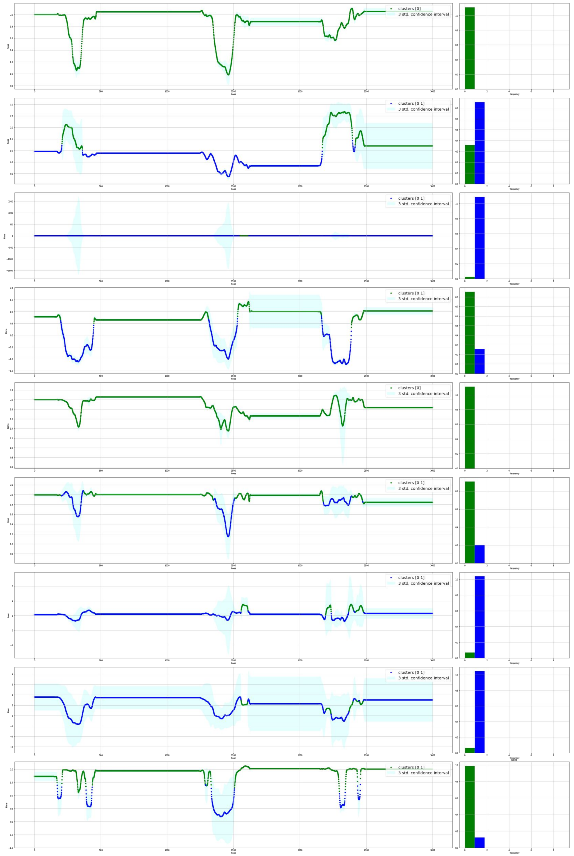


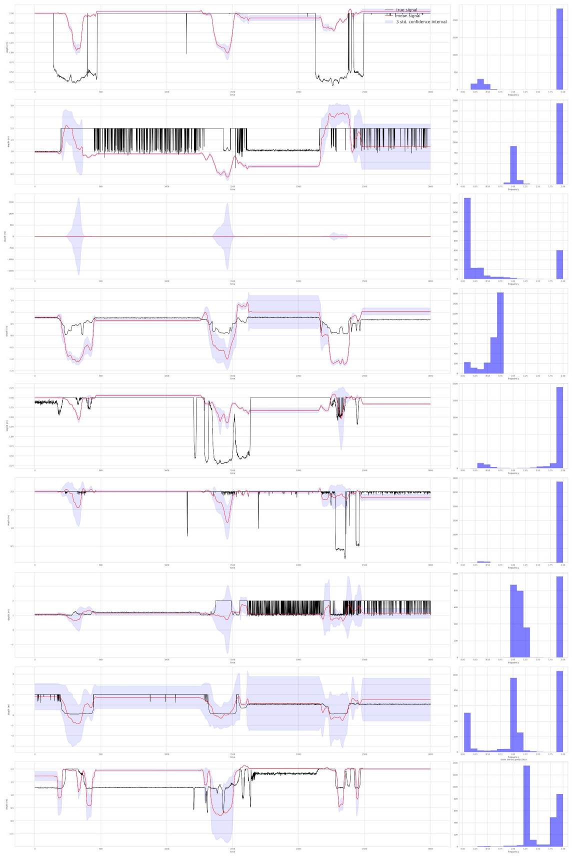
10 sample x



1 sample x v

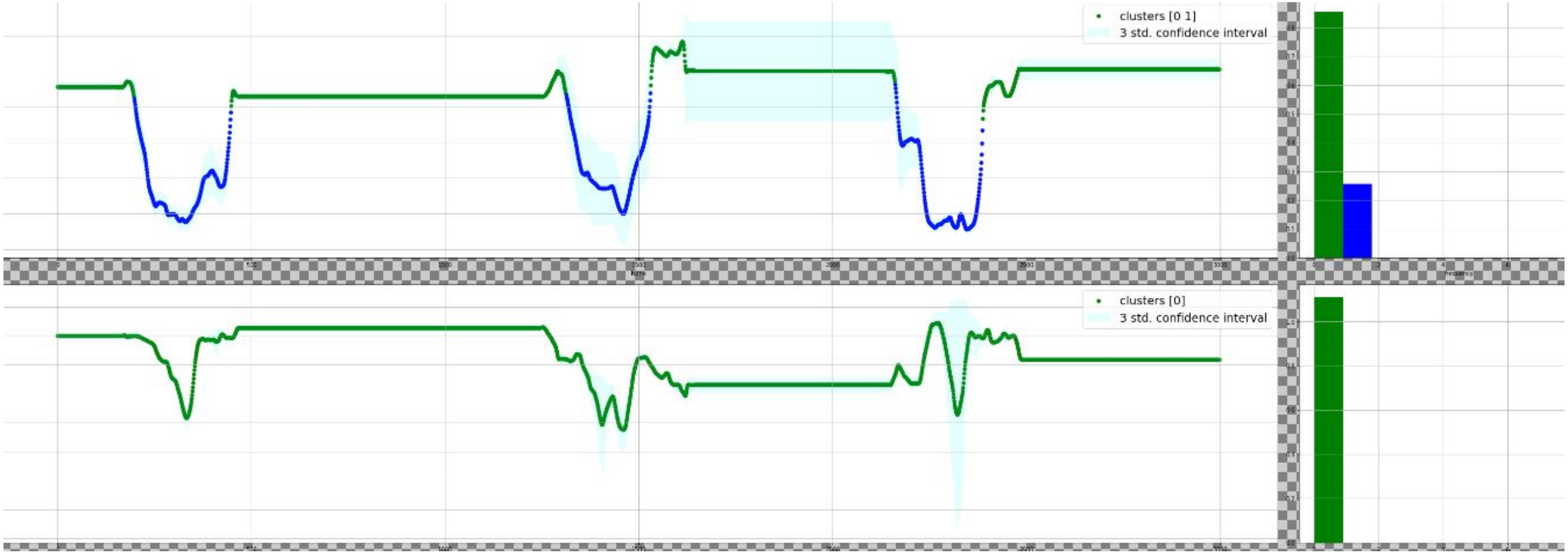
Result Selector for Clustering





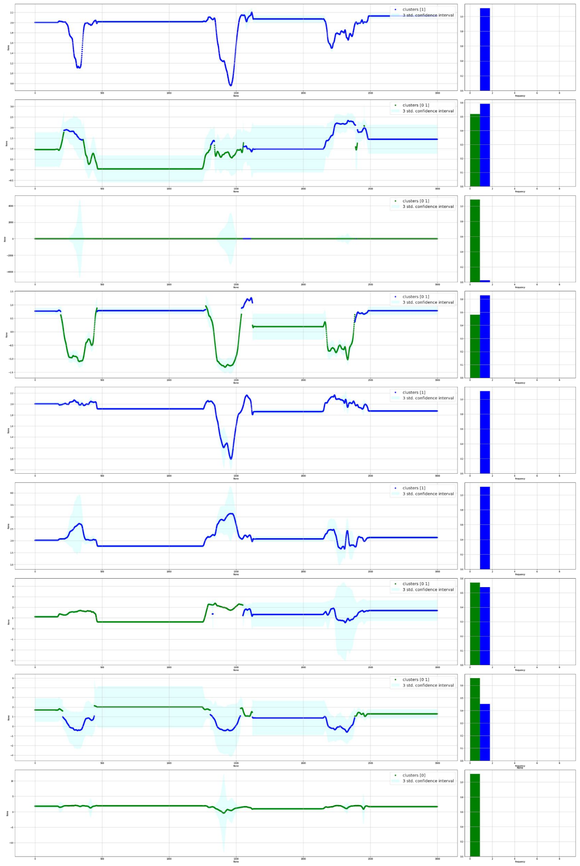
Result Selector for Clustering

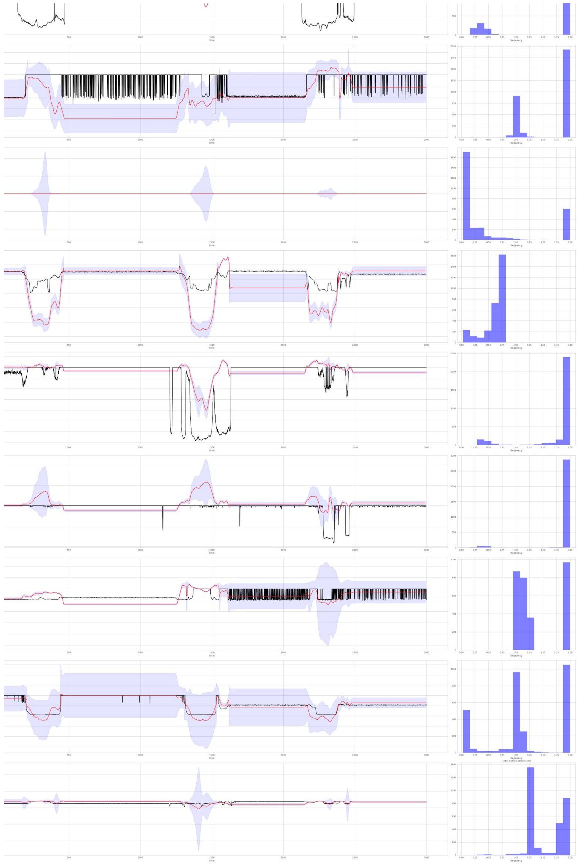
Lidar 3 we know gt



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

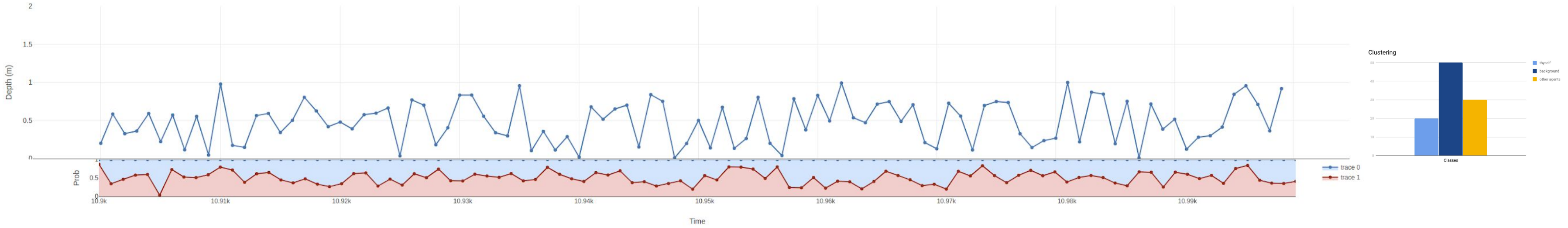
Result GMM for clustering



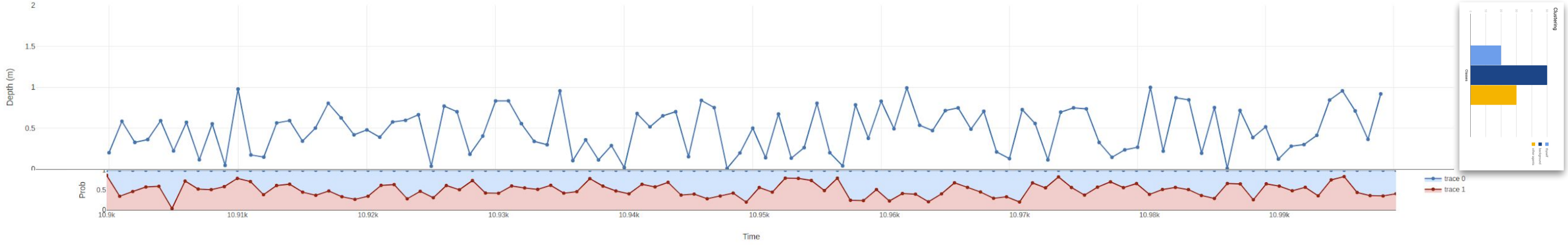


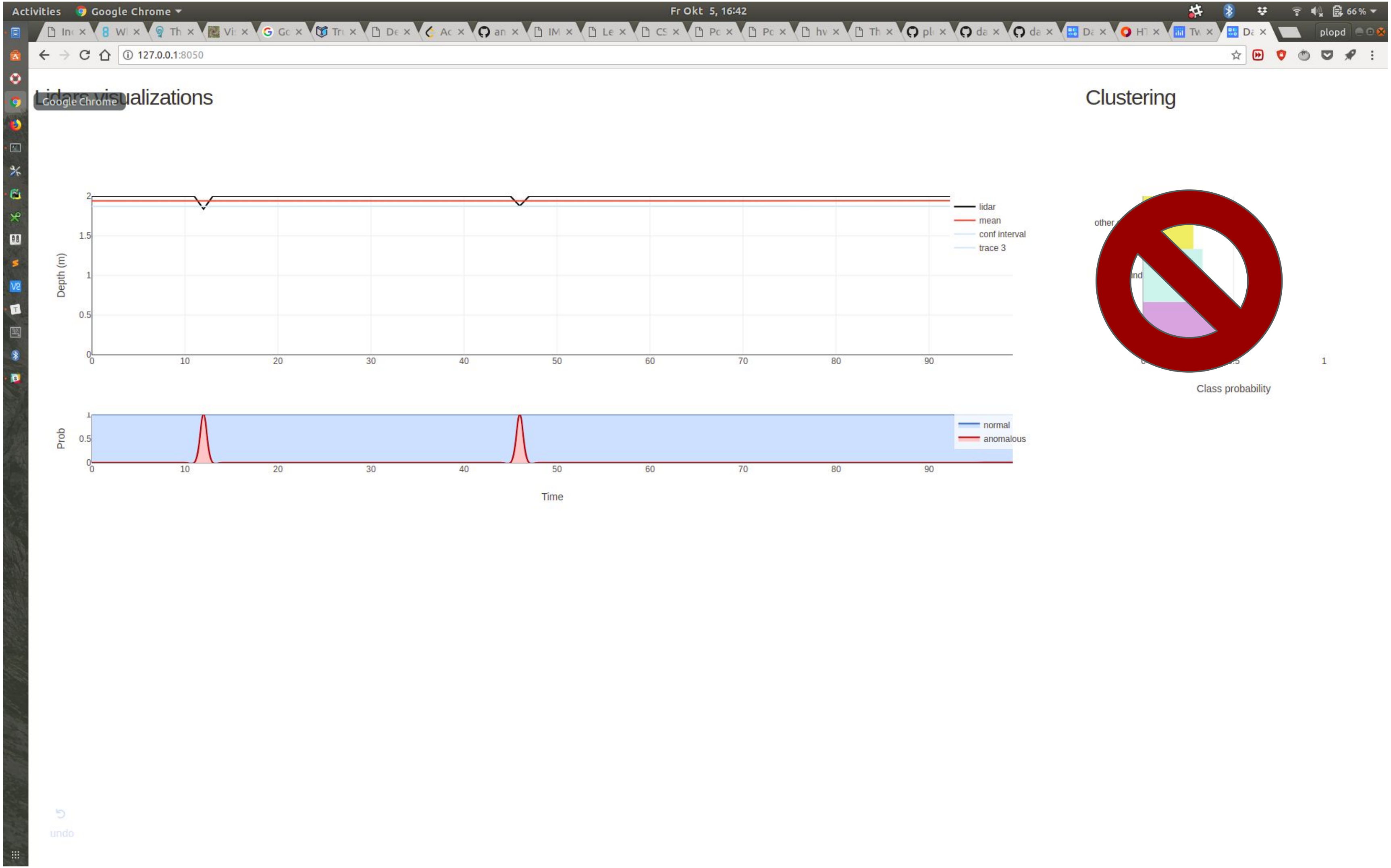
About demo...More Work in Progress!

Lidars visualizations



Lidars visualizations





Activities
Google Chrome
Fr Okt 5, 16:42
66 %
plopd
127.0.0.1:8050
Google Chrome
Lidars visualizations
Clustering
Depth (m)
lidar
mean
conf interval
trace 3
Prob
normal
anomalous
Time
other
Class probability
undo