Università di Genova
DIBRIS DIPARTIMENTO
DI INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Exam
Machine Learning and Data Analysis

Robotics Engineering

Alberto Di Donna, Andrea Chiappe e Fabio Guelfi

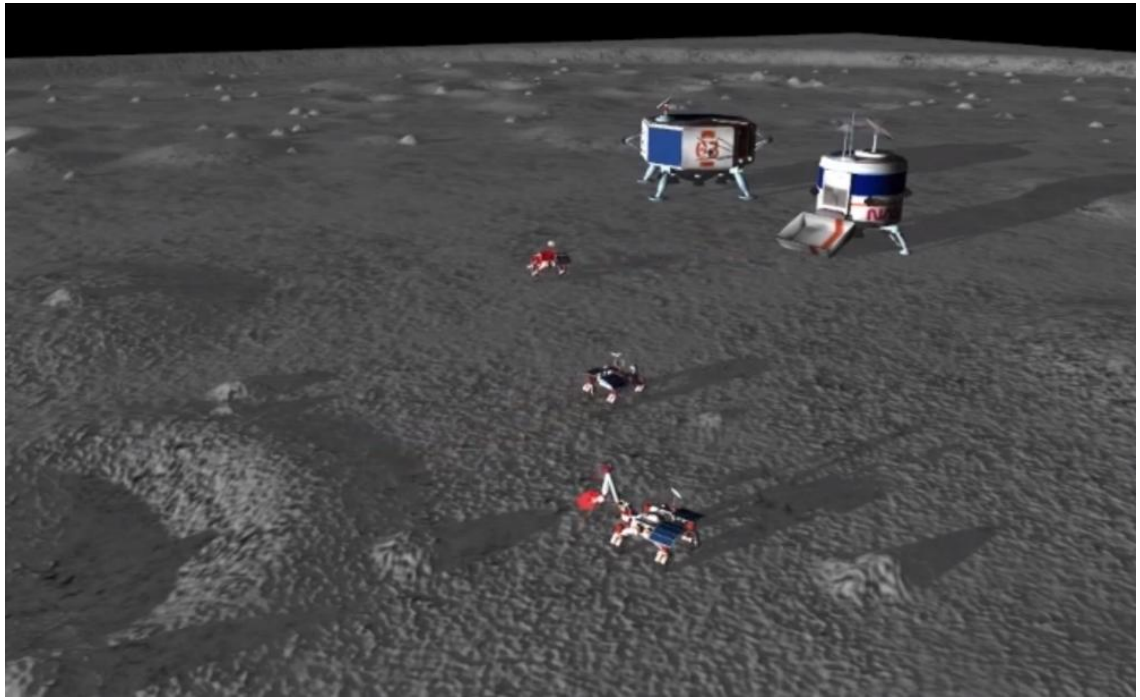
Università di Genova
DIBRIS DIPARTIMENTO
DI INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Dataset

Learned Pose Estimation (robot on the Moon)

Context

A simulated lunar surface, with lunar landers and lunar rovers.
The rovers must extract material from the soil and deliver it to a processing plant.

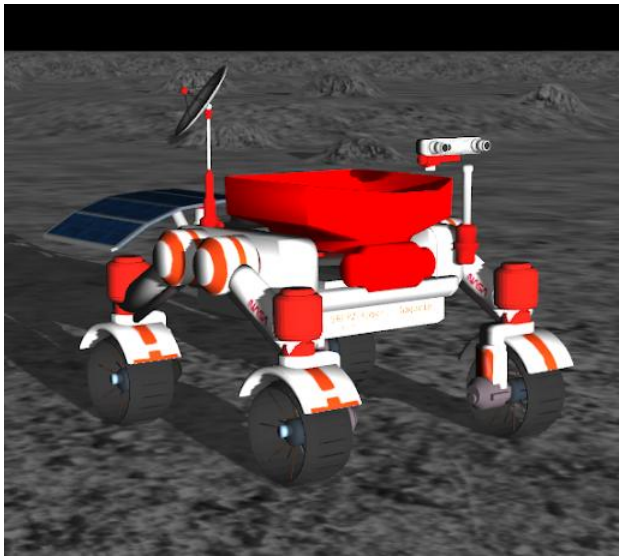


https://www.kaggle.com/datasets/louisburtz/learned-pose-estimation-robot-on-the-moon

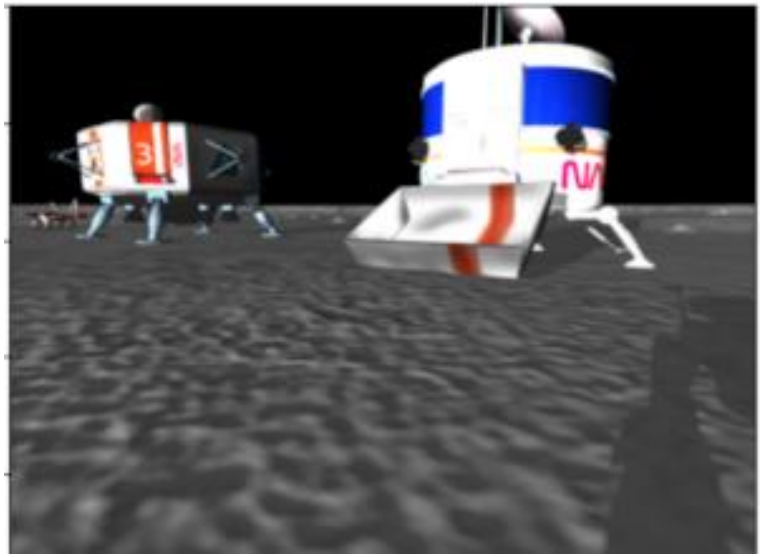
Learned Pose Estimation (robot on the Moon)

Goal

Estimate the pose of a Hauler rover's camera with respect to a known target object



Hauler Rover



Processing Plant Lander

https://www.kaggle.com/datasets/louisburtz/learned-pose-estimation-robot-on-the-moon

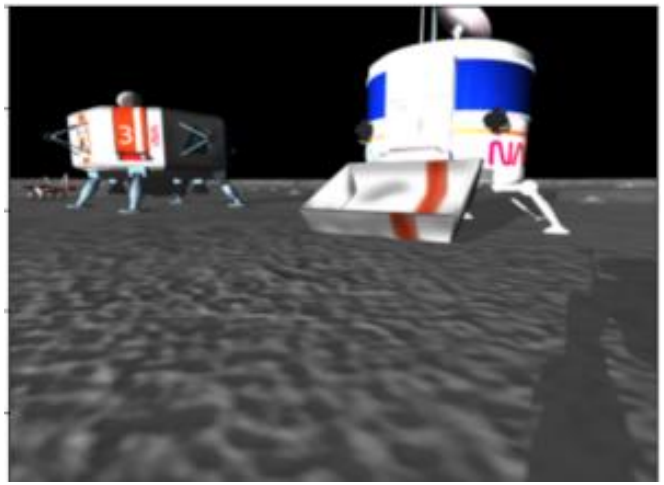
Learned Pose Estimation (robot on the Moon)

Data

- 4159 of RGBD images
- Gazebo ground truth (relative position and orientation)

Inference

- Input: single image (480x640x4)
- Output: relative position and orientation



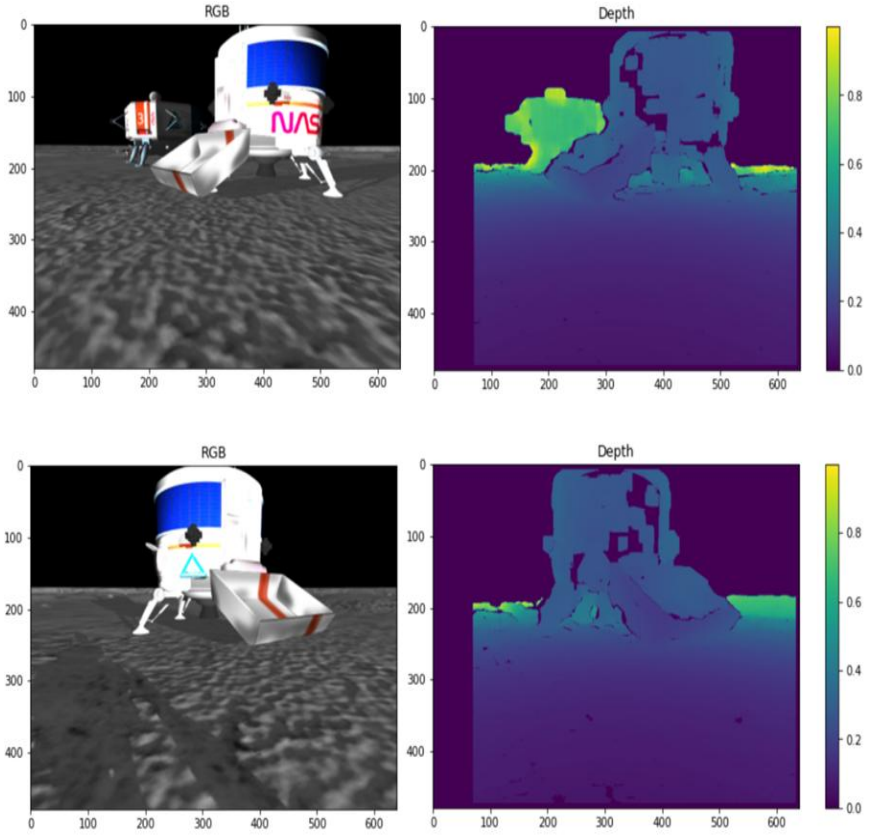
d = 8.36 m
theta = -47.9 deg,
yaw = 12.5 deg

https://www.kaggle.com/datasets/louisburtz/learned-pose-estimation-robot-on-the-moon

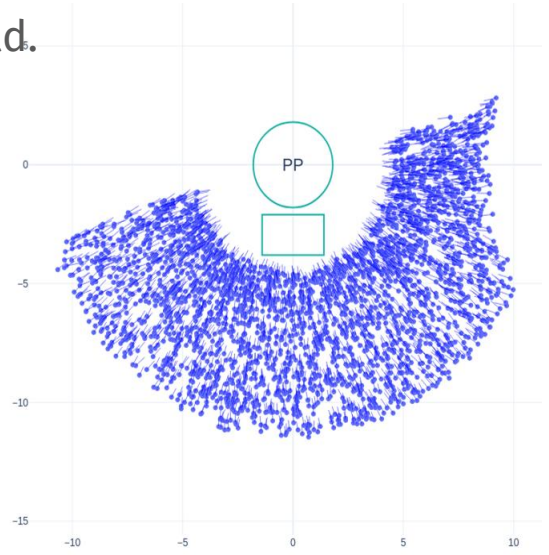
Learned Pose Estimation (robot on the Moon)

Data

an example of data... (batch!! 32x6 = 192)



- **Distance**: distance from target
- **Theta**: angle of object w.r.t. camera
- **Yaw**: orientation of the rover w.r.t. world.

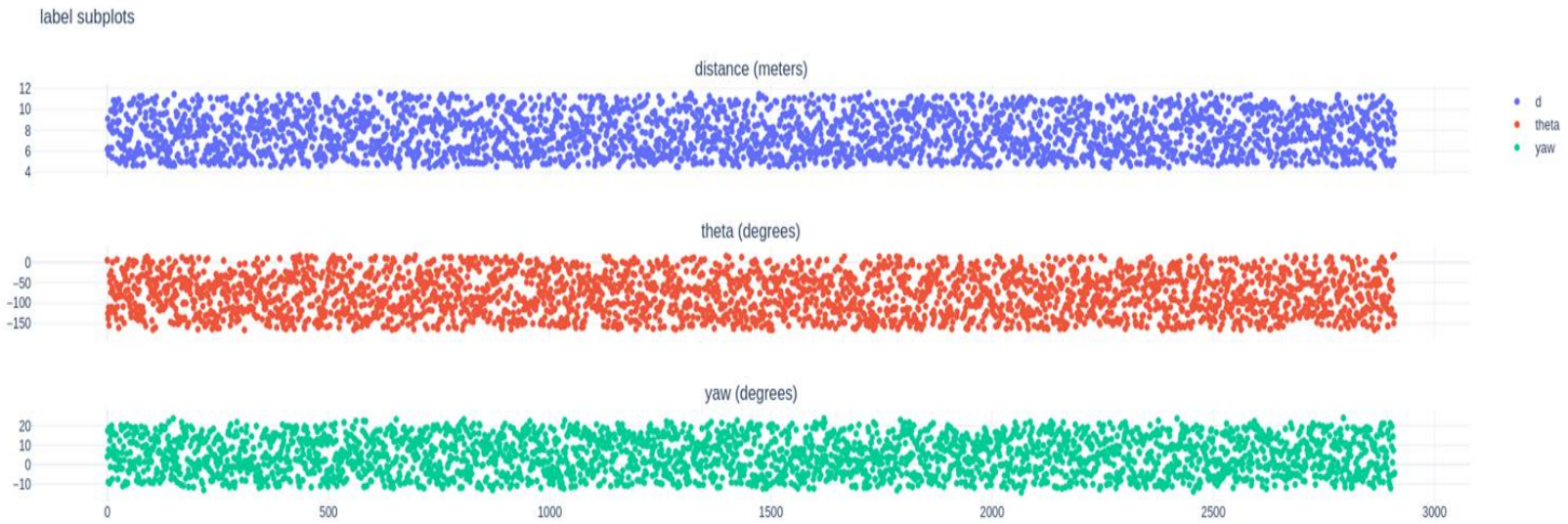


https://www.kaggle.com/datasets/louisburtz/learned-pose-estimation-robot-on-the-moon

Learned Pose Estimation (robot on the Moon)

Data (labels)

statistics: training set distribution (2911 points, 70% of dataset)

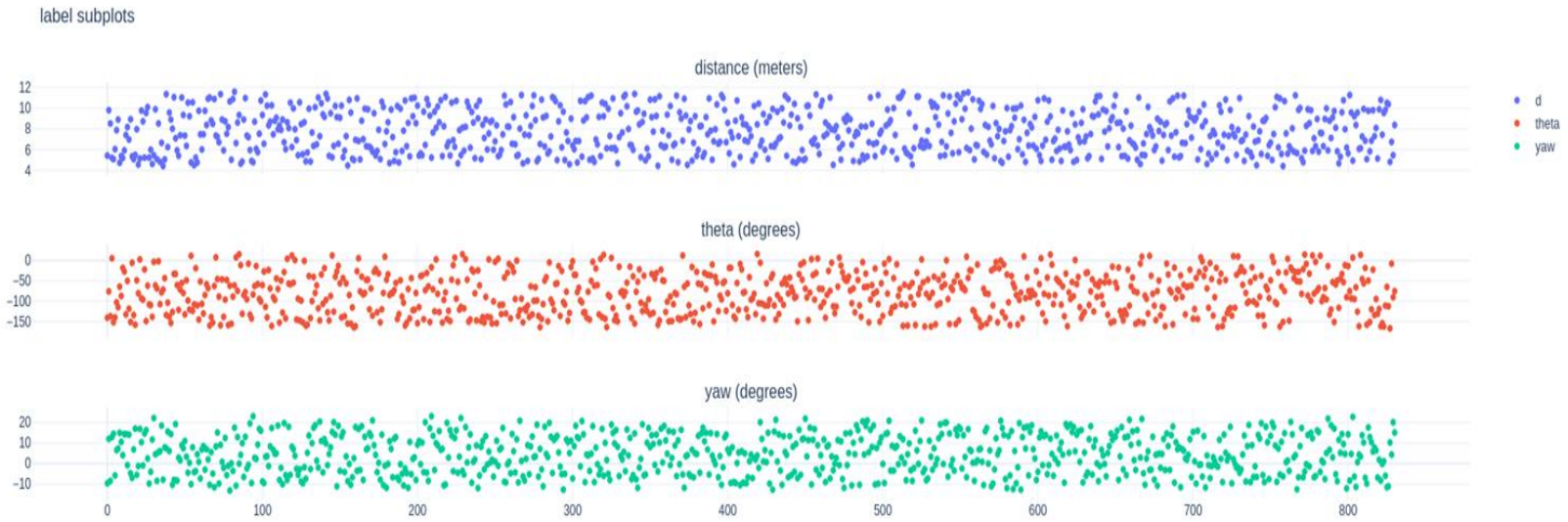


https://www.kaggle.com/datasets/louisburtz/learned-pose-estimation-robot-on-the-moon

Learned Pose Estimation (robot on the Moon)

Data

statistics: validation set distribution (831 points, 20% of dataset)

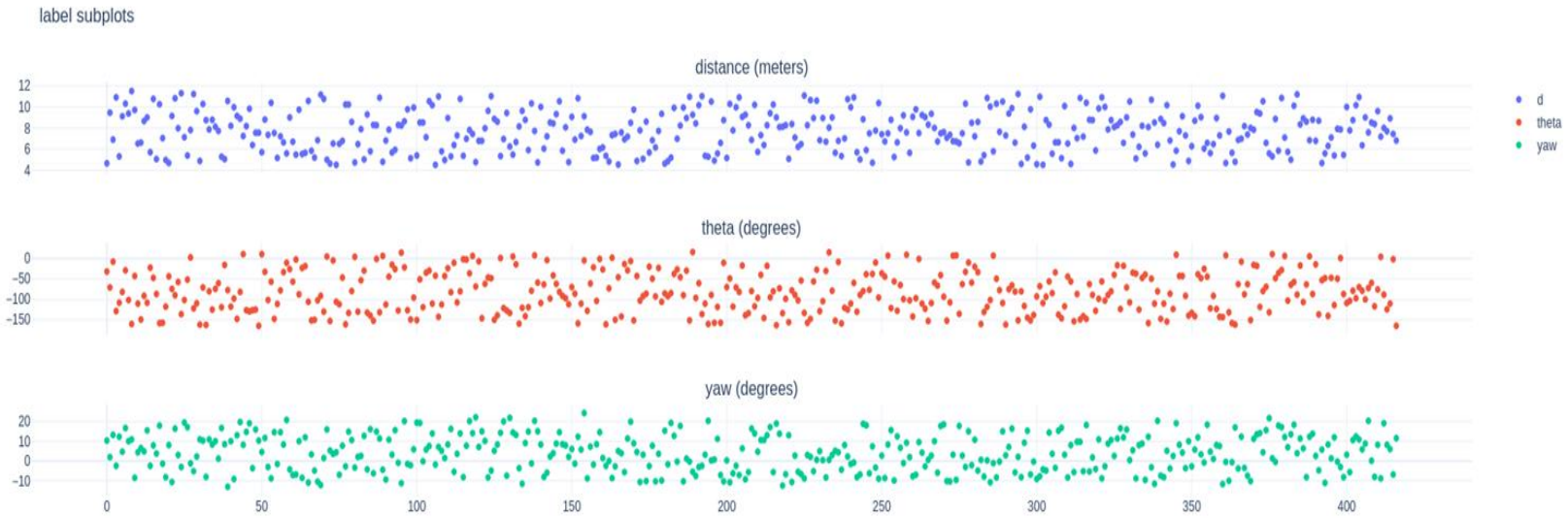


https://www.kaggle.com/datasets/louisburtz/learned-pose-estimation-robot-on-the-moon

Learned Pose Estimation (robot on the Moon)

Data

statistics: test set distribution (417 points, 10% of dataset)



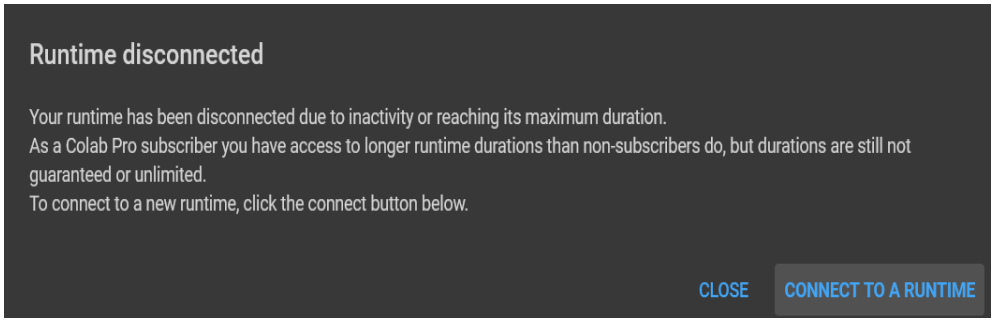
We can assume the tree sets are iid.

https://www.kaggle.com/datasets/louisburtz/learned-pose-estimation-robot-on-the-moon

Reduced Dataset

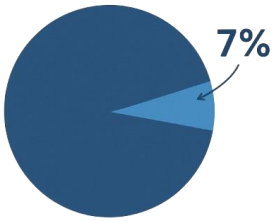
! Problem !

Execution required too much time and resources



We utilize only 1/15 of dataset

This solution speed up the fit phase allowing us to try different configuration and parameters.



Training, Test and Validation set are then reduceded in proportion

Università di Genova
DIBRIS DIPARTIMENTO
DI INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Model

Learned Pose Estimation (robot on the Moon)

Libraries

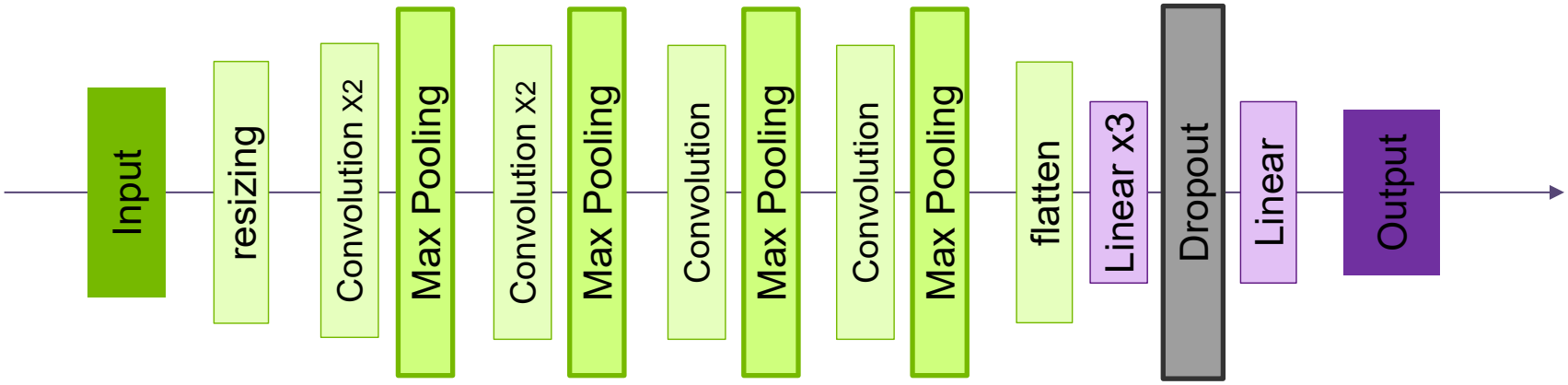
Open-source library for training and inference of neural networks.

High-level API for TensorFlow

User-friendly interface to develop the NN

Learned Pose Estimation (robot on the Moon)

Layers



Convolution layer:
- Conv2D
- Kernel size = 3x3
- Activation Function: relu

Max Pooling:
- Pooling size = 32, 64, 128

Learned Pose Estimation (robot on the Moon)

Layer Shape

Layer (type)	Output Shape	Param #
resizing (Resizing)	(None, 120, 160, 4)	0
conv2d (Conv2D)	(None, 120, 160, 32)	1,184
conv2d_1 (Conv2D)	(None, 120, 160, 32)	9,248
max_pooling2d (MaxPooling2D)	(None, 60, 80, 32)	0
conv2d_2 (Conv2D)	(None, 60, 80, 64)	18,496
conv2d_3 (Conv2D)	(None, 60, 80, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 30, 40, 64)	0
conv2d_4 (Conv2D)	(None, 30, 40, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 15, 20, 128)	0
conv2d_5 (Conv2D)	(None, 15, 20, 128)	147,584
max_pooling2d_3 (MaxPooling2D)	(None, 7, 10, 128)	0
flatten (Flatten)	(None, 8960)	0
dense (Dense)	(None, 6)	53,766
dense_1 (Dense)	(None, 24)	168
dense_2 (Dense)	(None, 24)	600
dropout (Dropout)	(None, 24)	0
dense_3 (Dense)	(None, 3)	75

Resizing → bring all images to the same size

Convolution → extracts relevant features through correlation between close pixels.

Max pooling → compresses information while retaining the most salient parts

Dropout → randomly turns off neurons during training to avoid overfitting and improve generalization.

Linear at the end → because it is regression: we want continuous real outputs, not limited by the activation functions

Activation Function → Relu

Learned Pose Estimation (robot on the Moon)

Loss function

Weighted average of MSE on distance, theta and yaw

```
def pose_loss(y_true, y_pred):
    pose_loss = \
        gamma * distance_loss(y_true, y_pred) +  \
        alpha * theta_loss(y_true, y_pred) + \
        beta  * orientation_loss(y_true, y_pred)
    return pose_loss
```

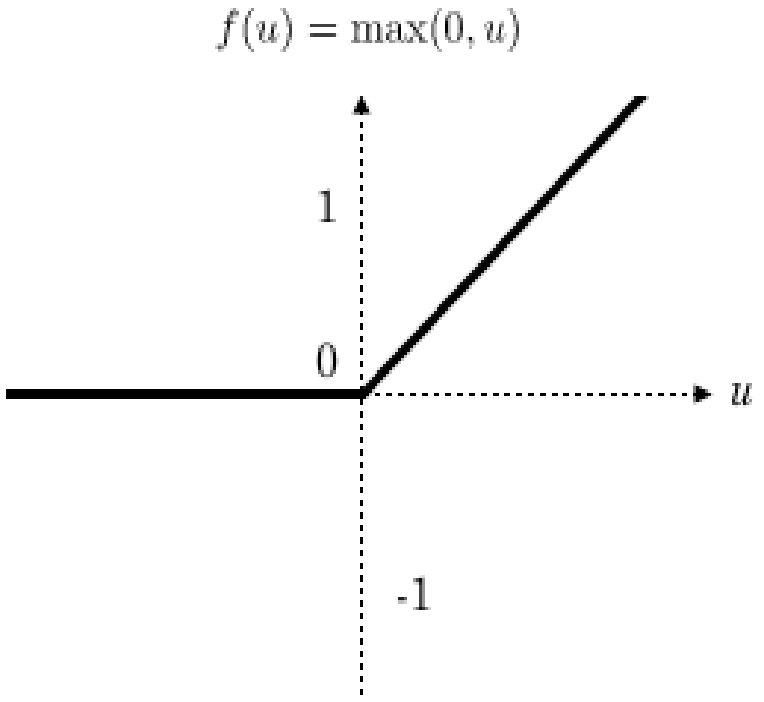
$$L(x) = \gamma \frac{1}{N}\sqrt{y^2 + y_p^2} + \alpha \frac{1}{N}\sqrt{\theta^2 + \theta_p^2} + \beta \frac{1}{N}\sqrt{\psi^2 + \psi_p^2}$$

Most of the work done is about find the best possible parameters to use into the loss function

Learned Pose Estimation (robot on the Moon)

Activation Function: RELU

- Rectified linear unit
- Gives **non linearity** to the net
- **Easy** and **efficent** function
- Is possible to learn more compelx pattern

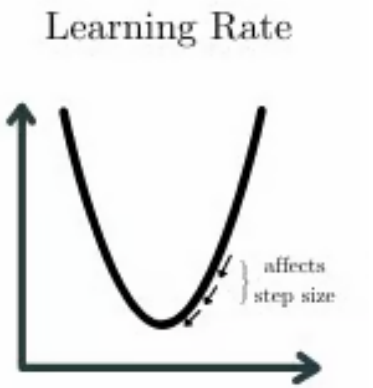


We didn’t need more complex function so, ReLU was our best possible choice

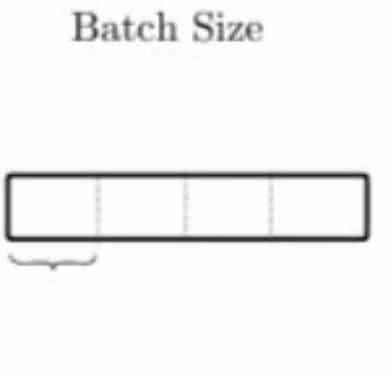
Learned Pose Estimation (robot on the Moon)

Learning rate & Batch size

Learning Rate = 0.001 & Batch size = 32



The LR value was chosen because if it were too large it would cause oscillations, while if it were too small the training would become too slow.



The batch size value offered GPU computational efficiency, even if not big, and ensured better generalization than higher values

These values were our best compromise for the net

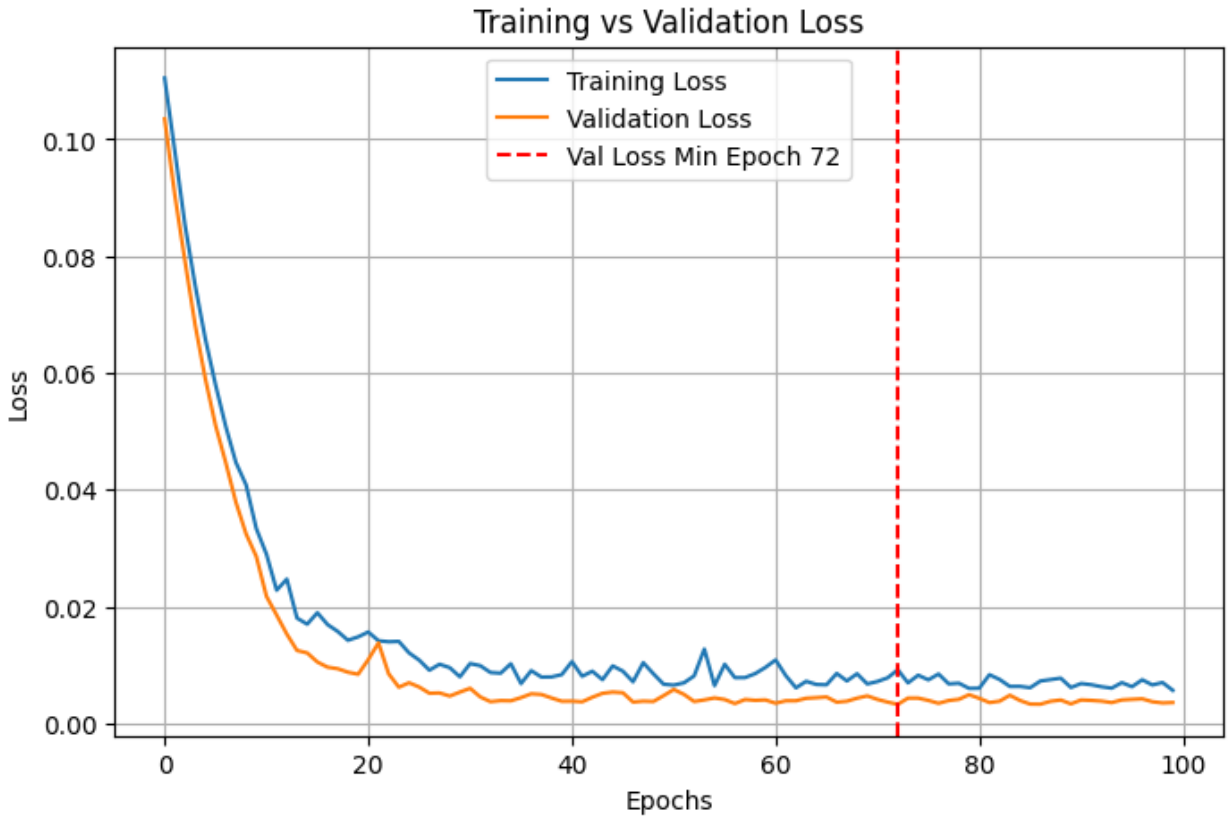
Università di Genova
DIBRIS DIPARTIMENTO
DI INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Results

Net Parameters (1)

$\gamma = 1/3;\ \beta = 1/3;\ \alpha = 1/3$

Epochs = 100

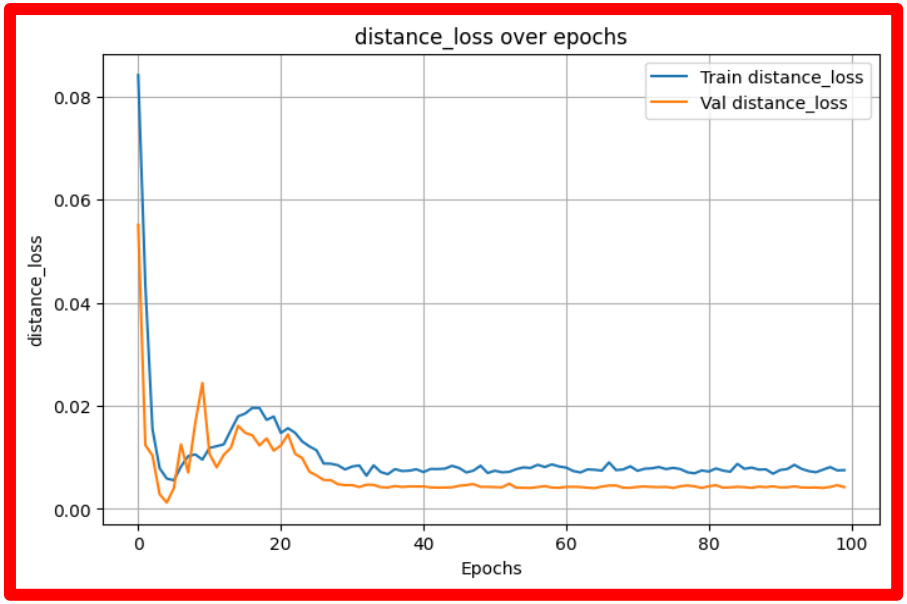


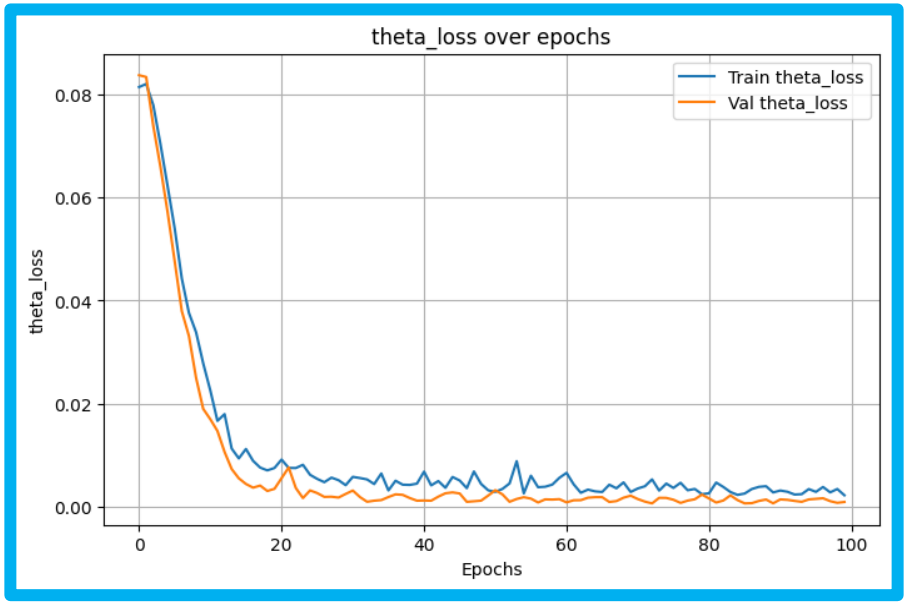
Net Parameters (1)

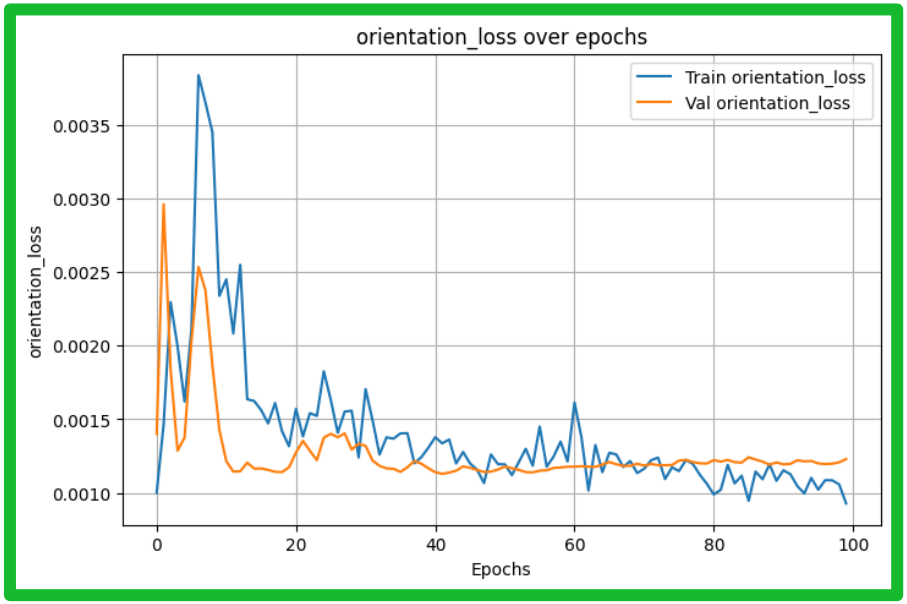
γ = 1/3; β = 1/3; α = 1/3

Epochs = 100

$$L(x) = \gamma \frac{1}{N}\sqrt{y^2 + {y_p}^2} + \alpha \frac{1}{N}\sqrt{\theta^2 + {\theta_p}^2} + \beta \frac{1}{N}\sqrt{\psi^2 + {\psi_p}^2}$$

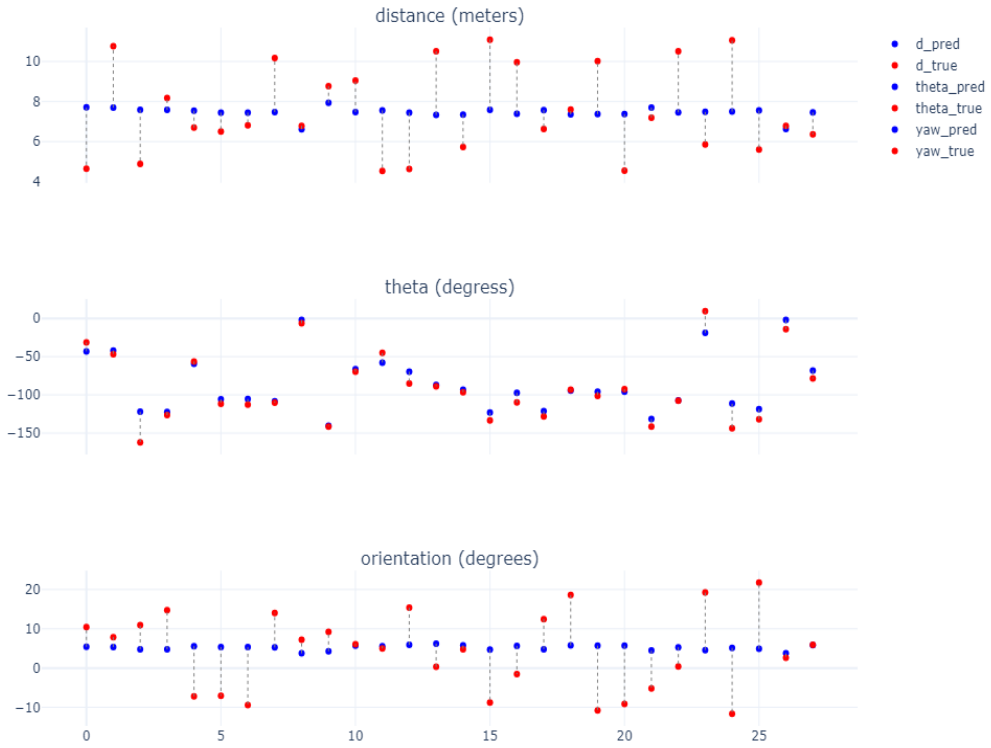






Predict vs Ground Truth

$\gamma = 1/3;\ \beta = 1/3;\ \alpha = 1/3$



The performance of the net is evaluated on Test set computing the **Mean absolute error** on distance, theta and yaw

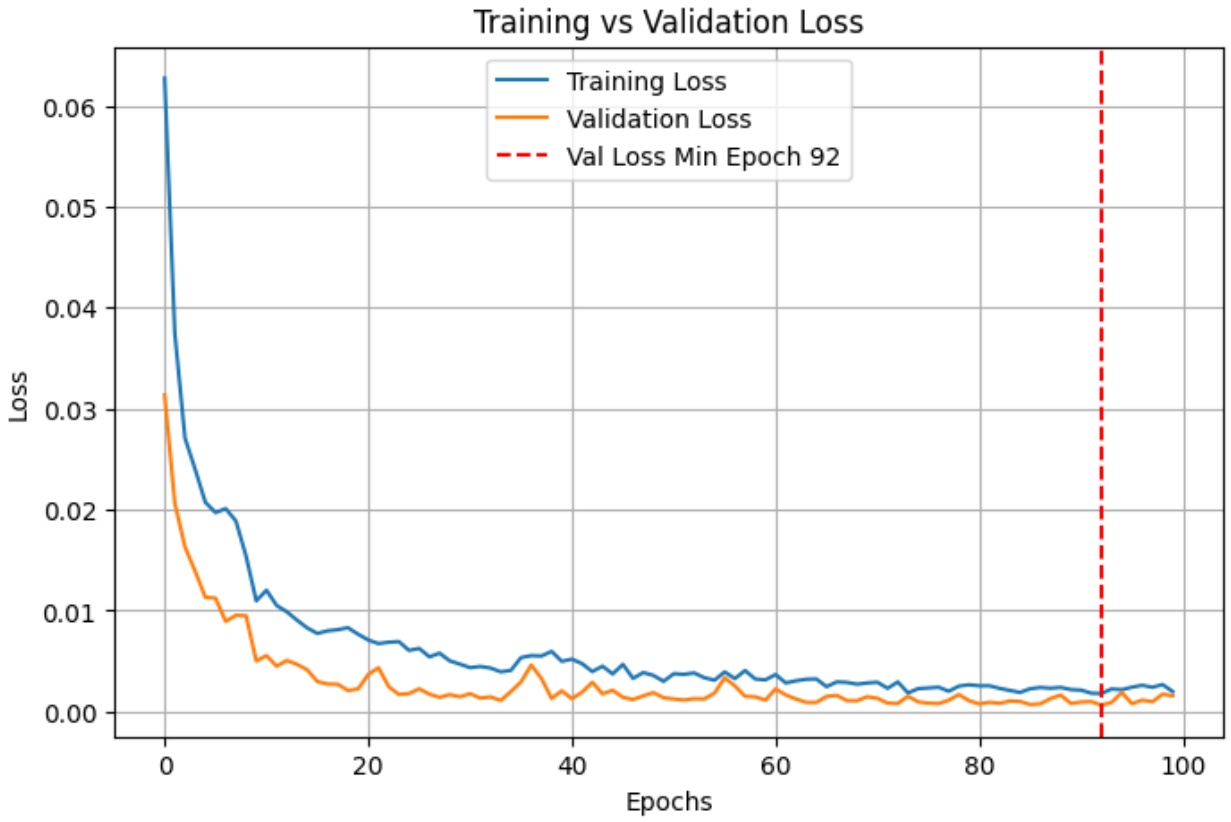
Results evaluation

- avg_position_diff = 1.903 meters
- avg_theta_diff = 8.3 degrees
- avg_orientation_diff = 8.3 degrees

Net Parameters (2)

$\gamma = 1/2;\ \beta = 1/4;\ \alpha = 1/4$

Epochs = 100

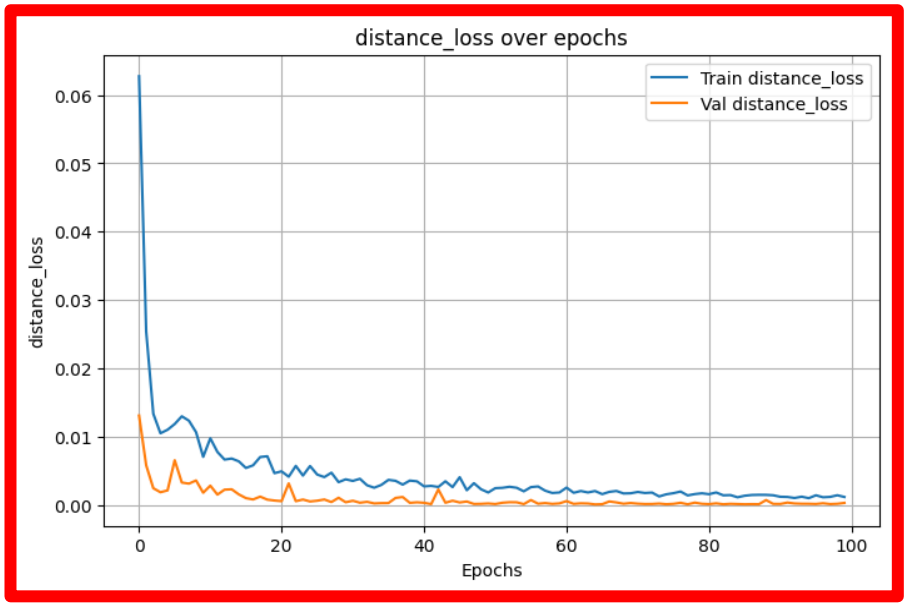


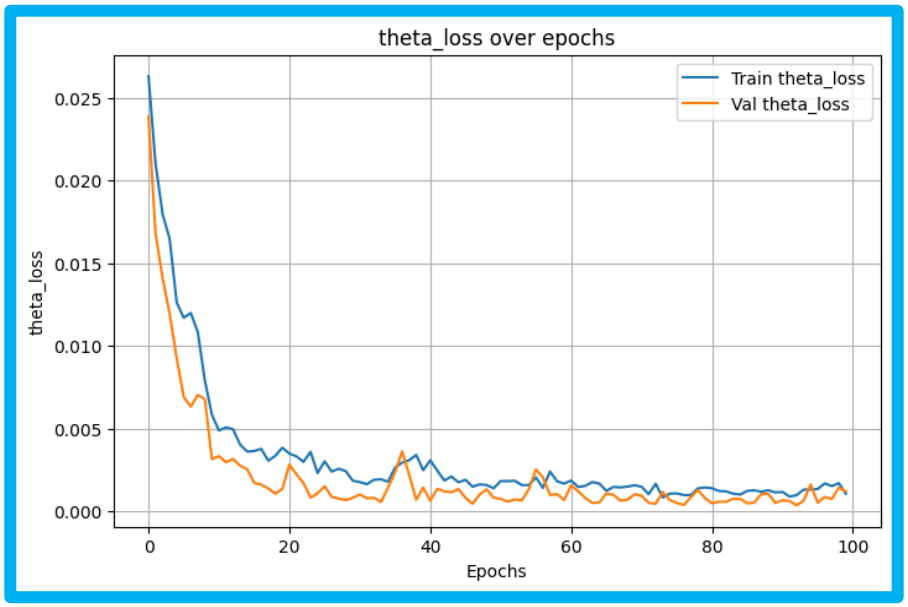
Net Parameters (2)

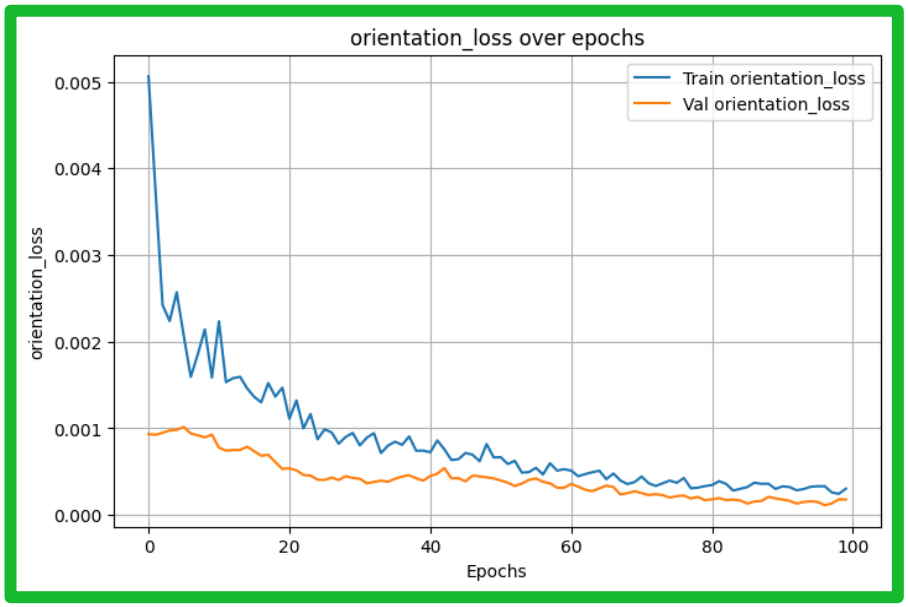
$\gamma = 1/2;\ \beta = 1/4;\ \alpha = 1/4$

Epochs = 100

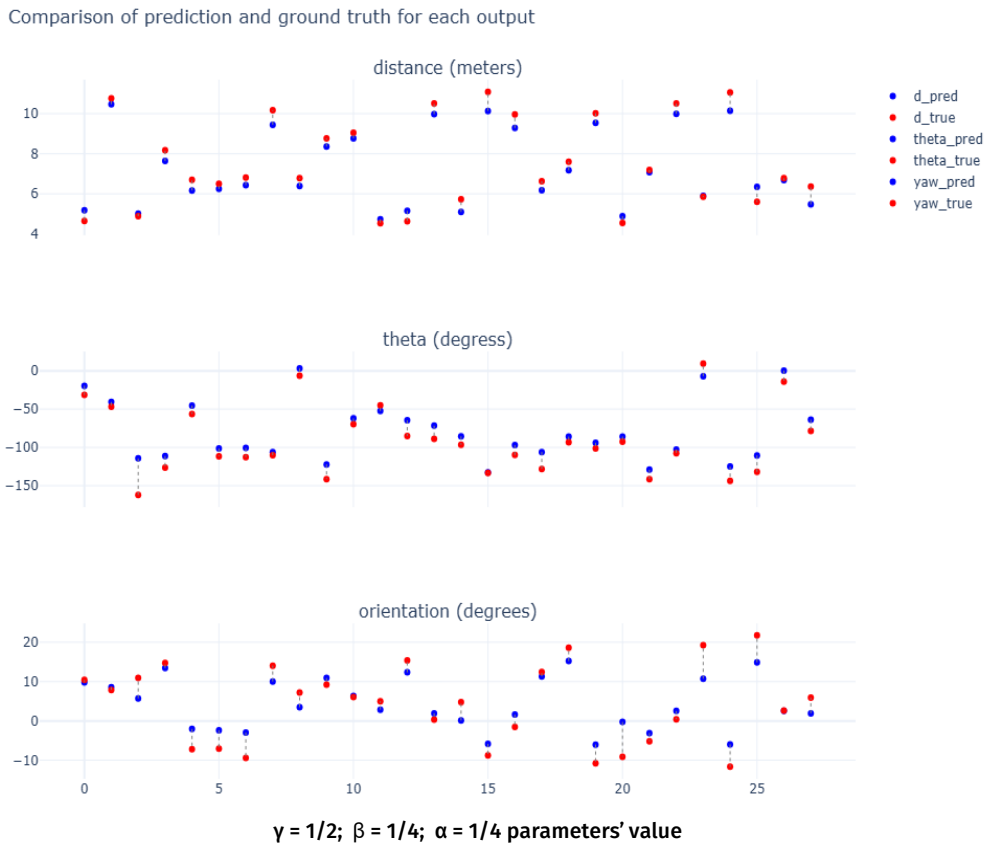
$$L(x) = \gamma \frac{1}{N}\sqrt{y^2 + y_p^2} + \alpha \frac{1}{N}\sqrt{\theta^2 + \theta_p^2} + \beta \frac{1}{N}\sqrt{\psi^2 + \psi_p^2}$$







Prediction vs Ground Truth

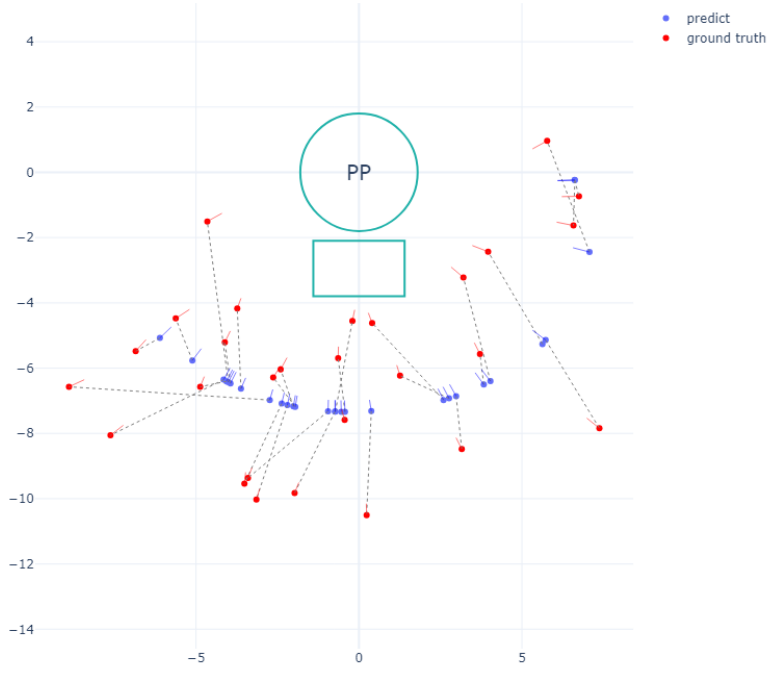


γ = 1/2; β = 1/4; α = 1/4 parameters' value

avg_position_diff = 0.464 meters
avg_theta_diff = 13.2 degrees
avg_orientation_diff = 3.5 degrees

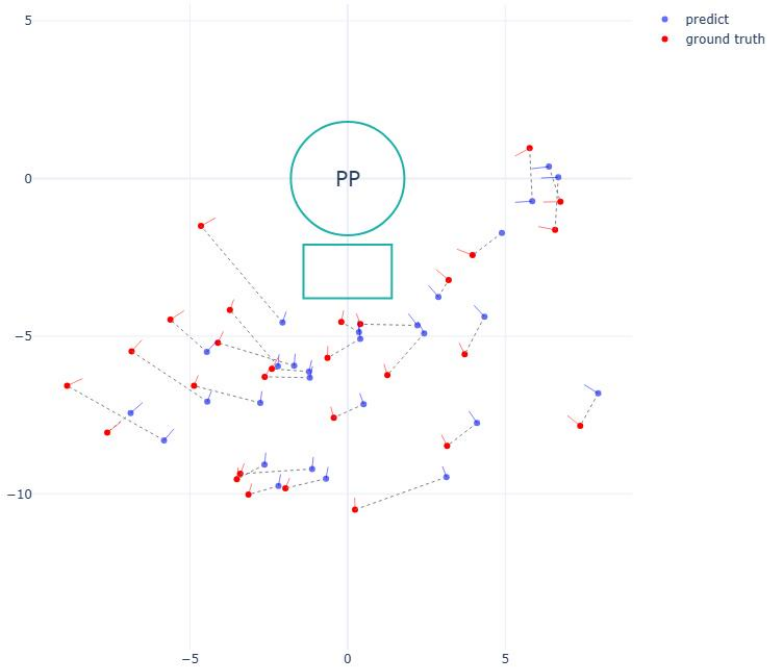
Comparison Parameters (1) vs (2)

γ = 1/3; β = 1/3; α = 1/3



avg_position_diff = 1.903 meters
avg_theta_diff = 8.3 degrees
avg_orientation_diff = 8.3 degrees

γ = 1/2; β = 1/4; α = 1/4



avg_position_diff = 0.464 meters
avg_theta_diff = 13.2 degrees
avg_orientation_diff = 3.5 degrees

UniGe

DIBRIS