Meeting 03/17/2020

Shuo Zhang

Exclude Bad Data

a) data of bad detection(tag not detected, coordinate system not parallel, …) and drop

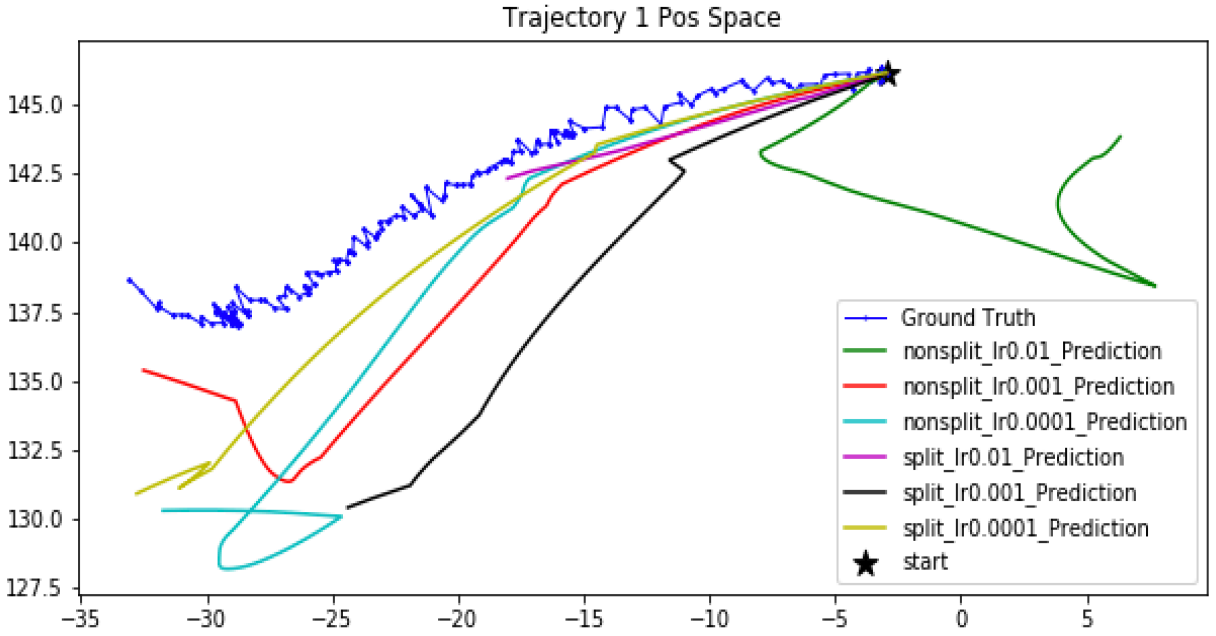
Information coming from raw data, due to bad visual detection

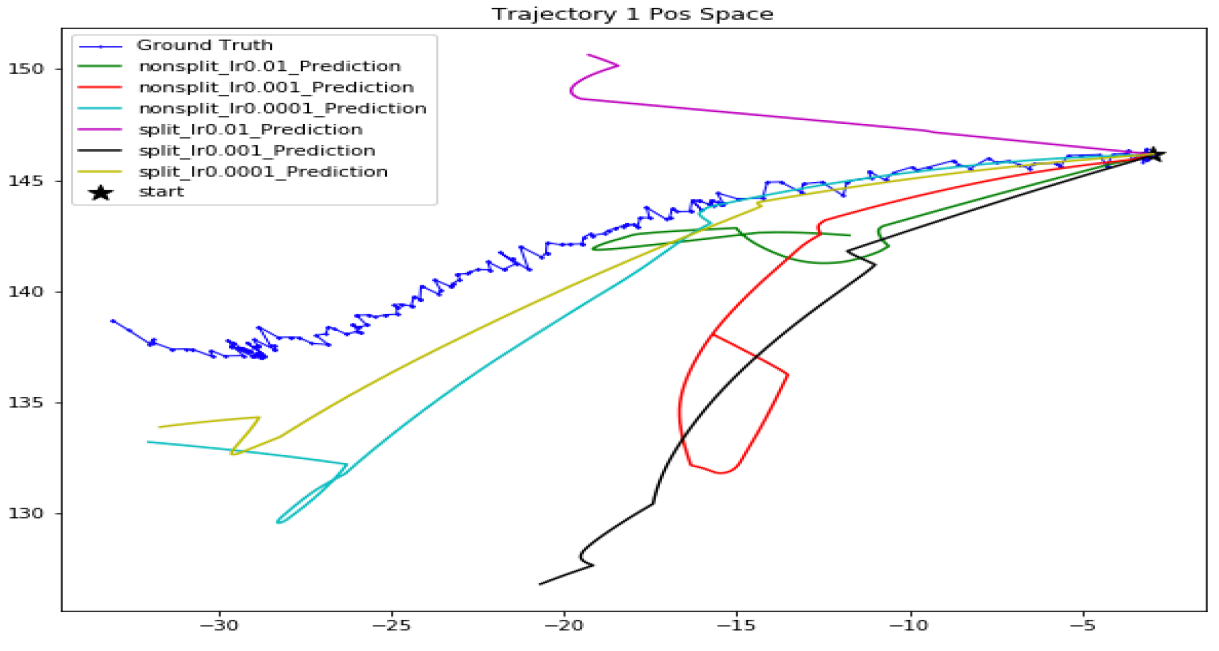
Calibration Matrix Computing & Dealing with Bad Missing Data Point

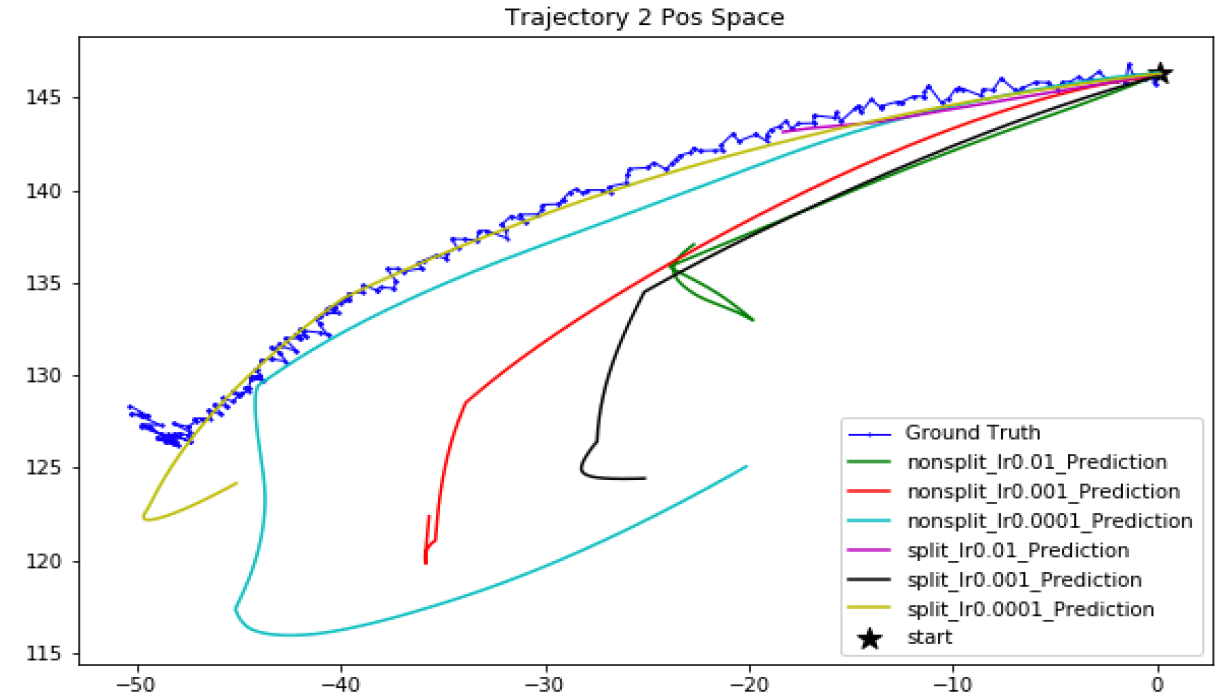
Should be excluded from both Calibration Matrix Computing and Training Data

Training results until beginning of March:

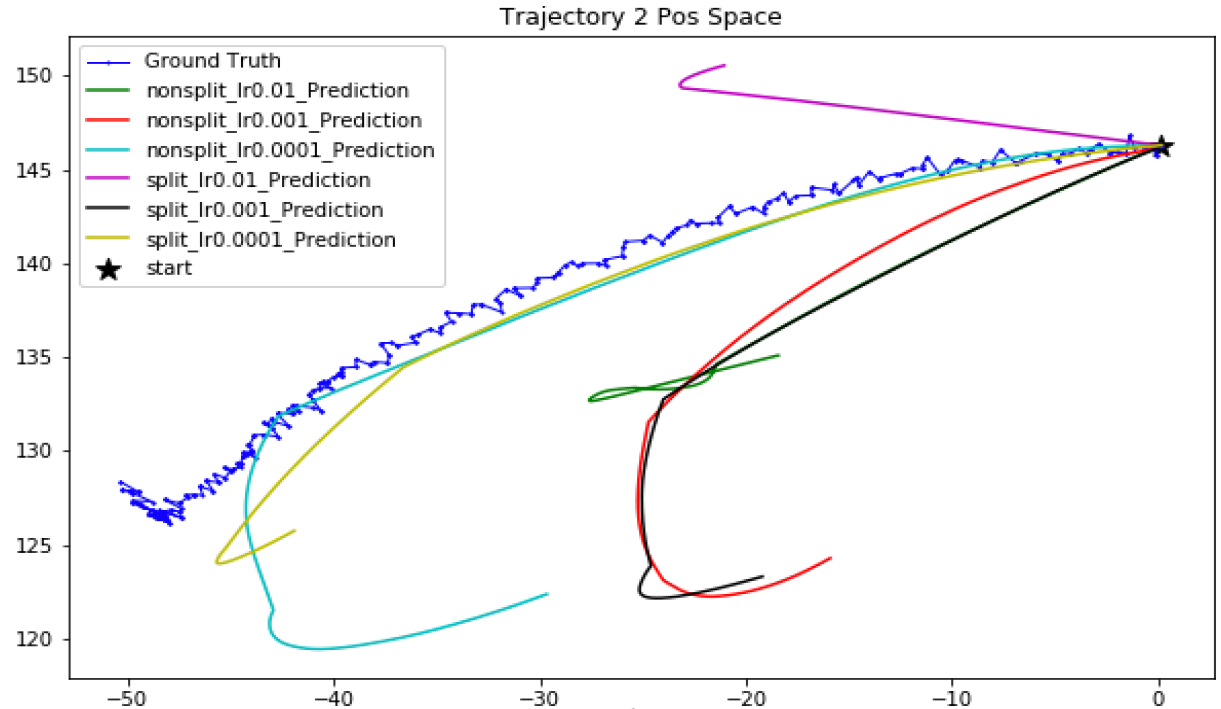
learning rate: [0.01, 0.001, 0.0001]; network: [split, nonsplit]; hidden nodes: [200 ,512]





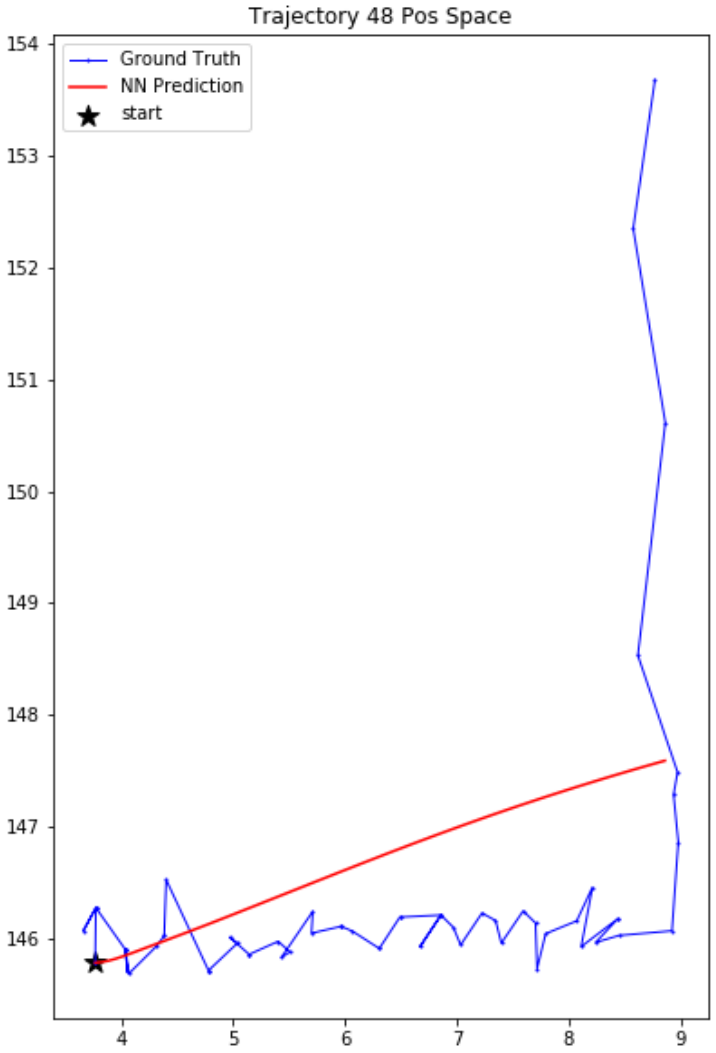


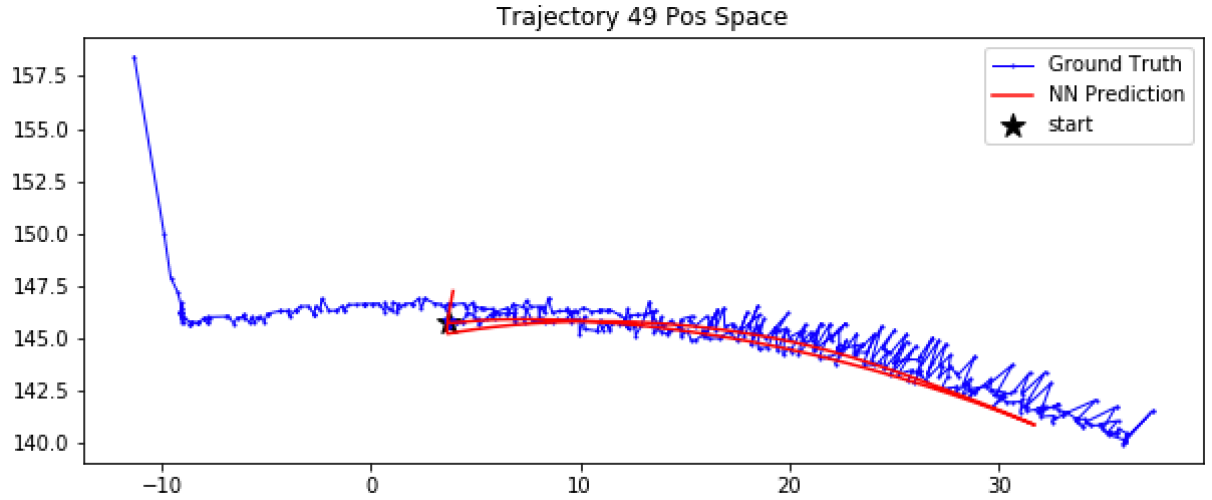
Nodes: 200

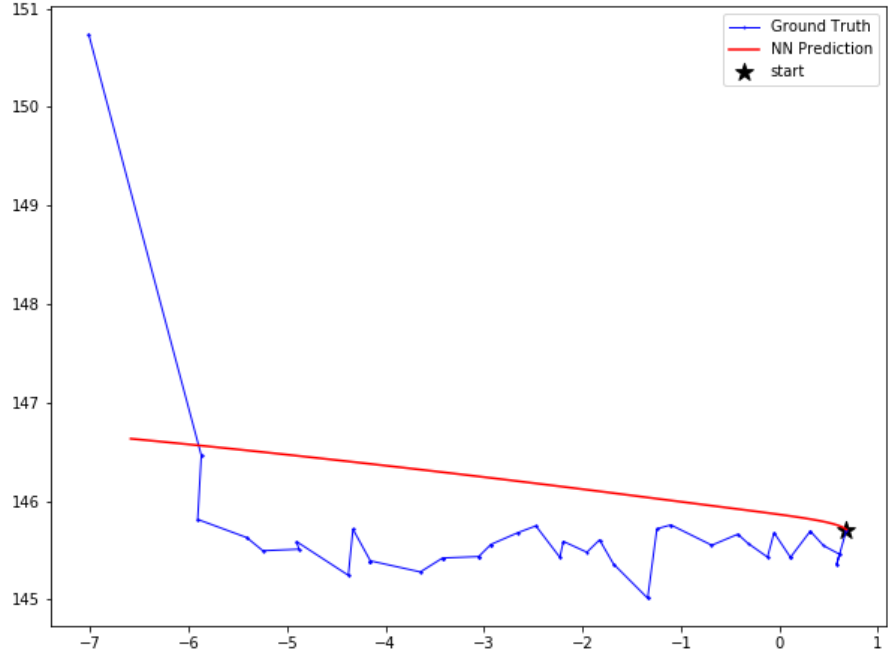


Nodes: 512

In most cases, predictions inaccurate. Data too messy!

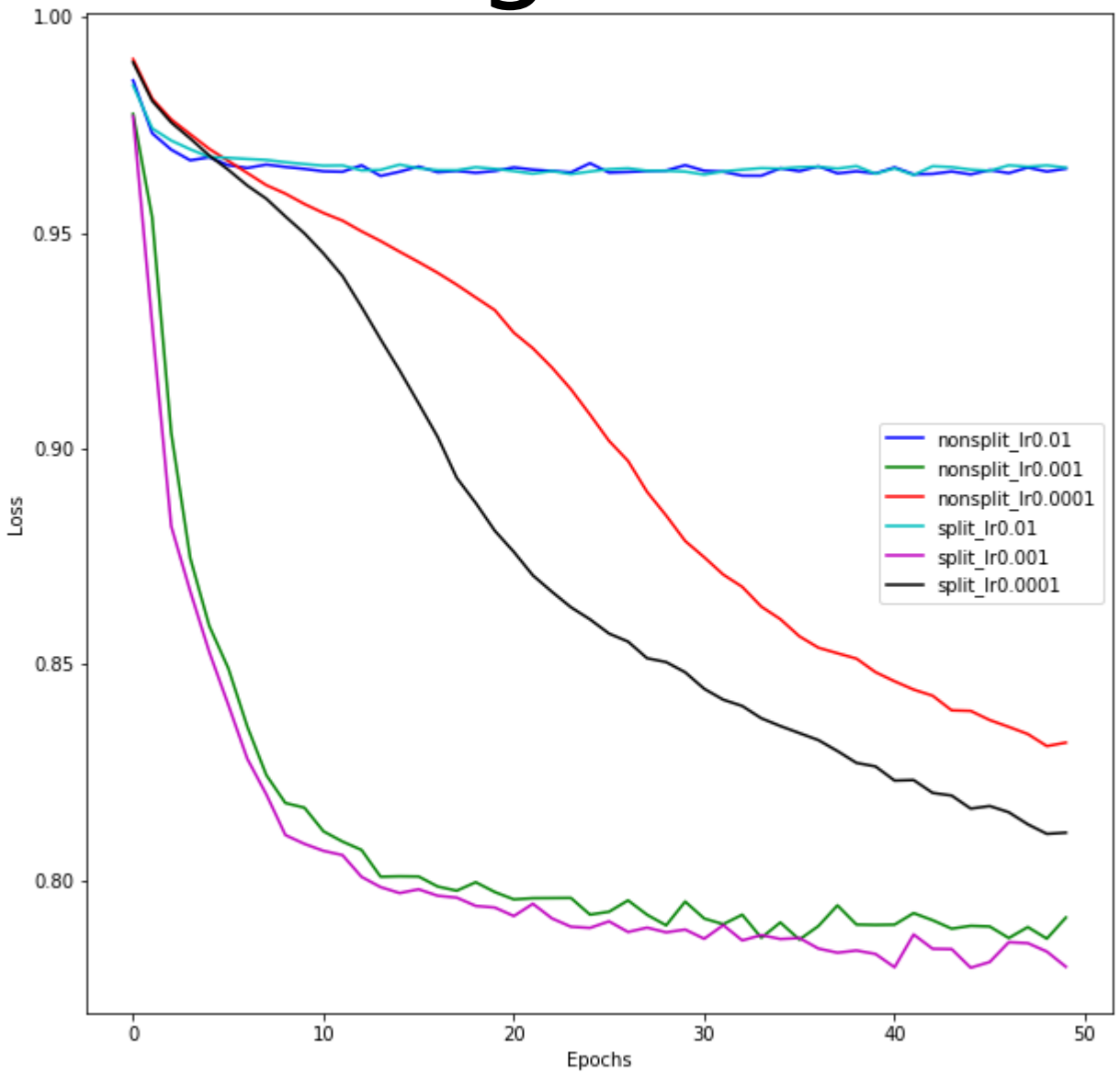




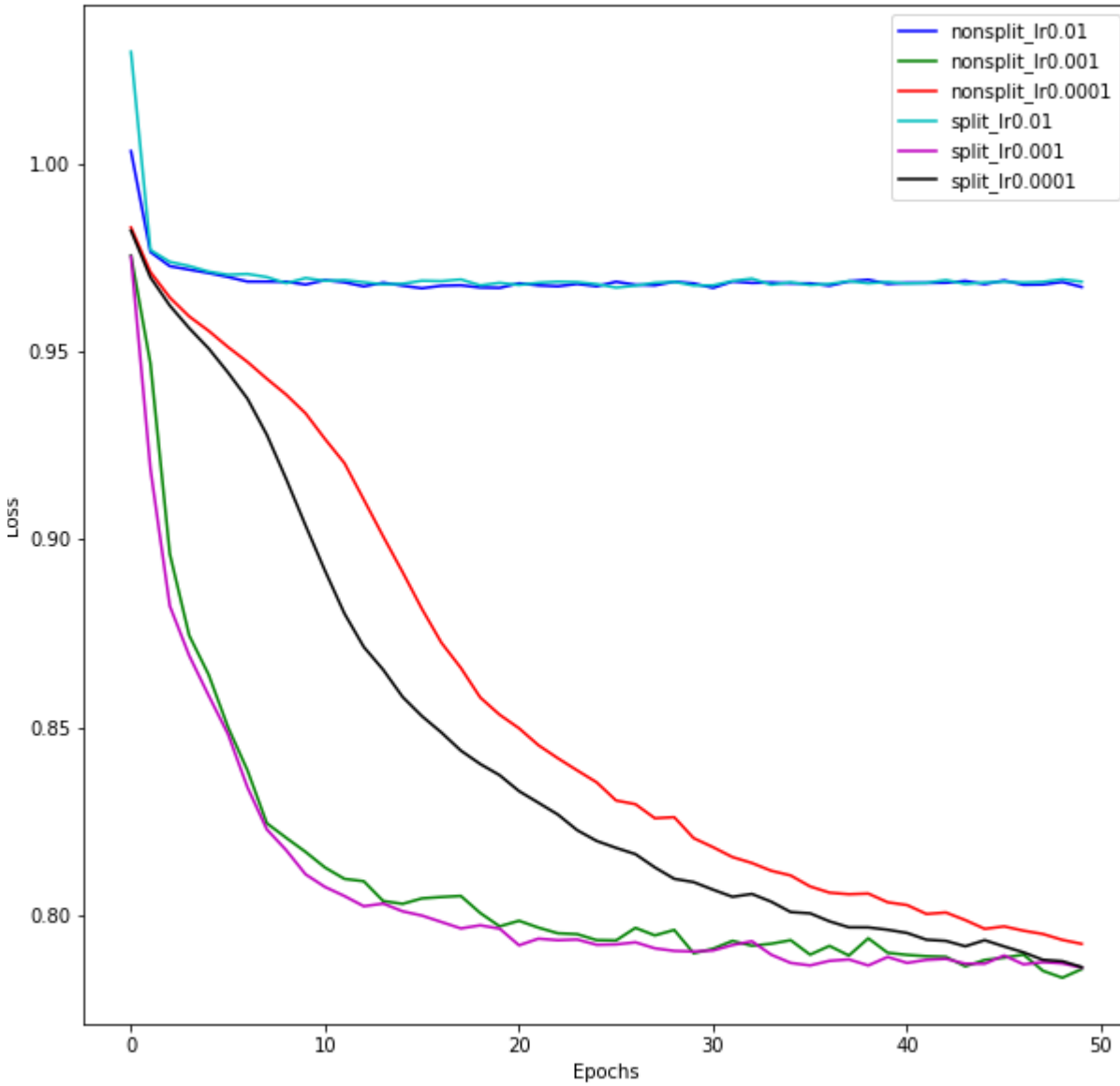


Some train trajectories prediction with models of 200 Nodes, Learning rate 0.0001

Training Error

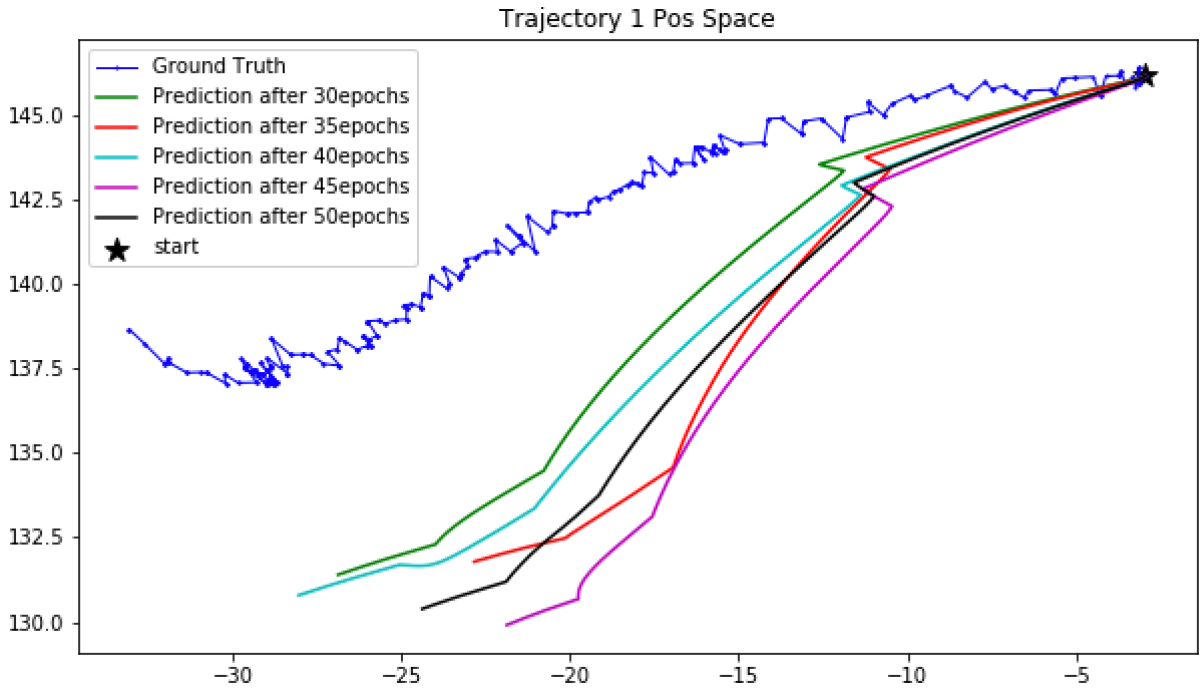


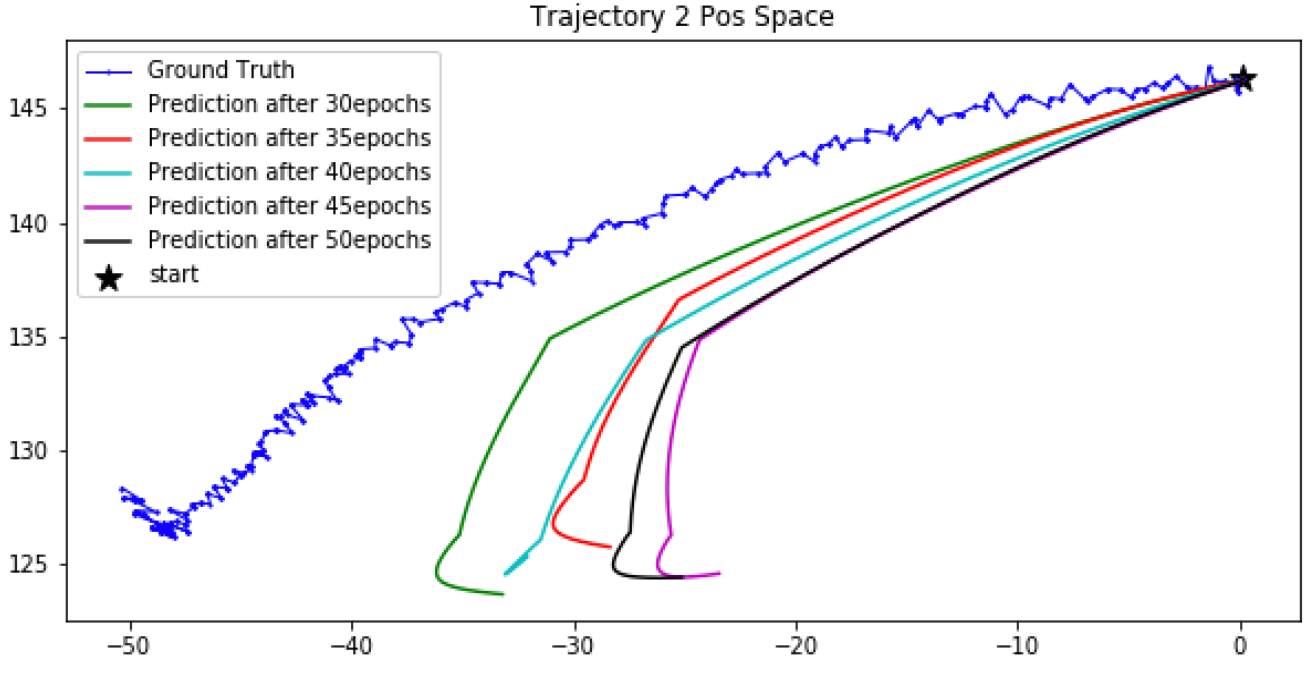
Nodes: 200



Nodes: 512

Overfit observed





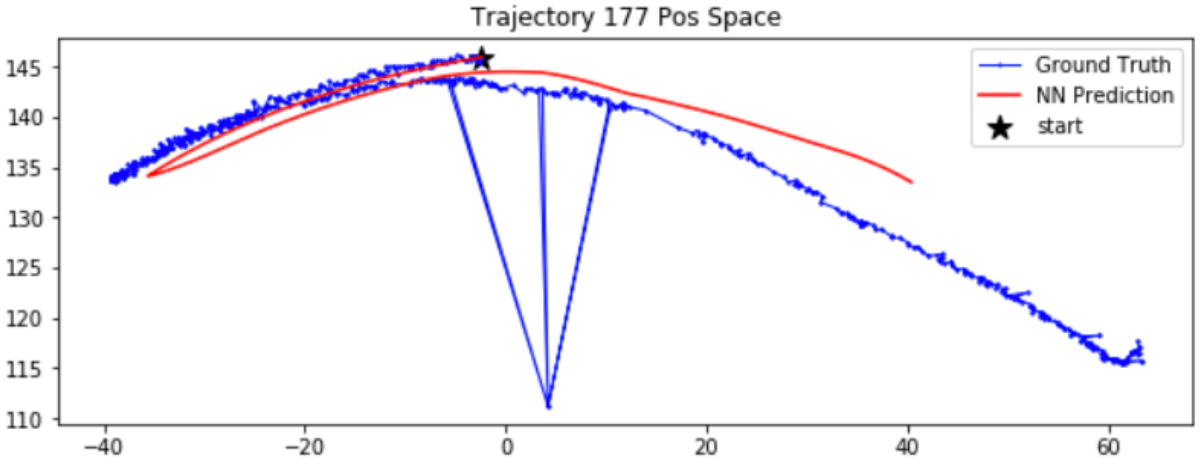
Split Model, 200 Nodes, Learning rate 0.001

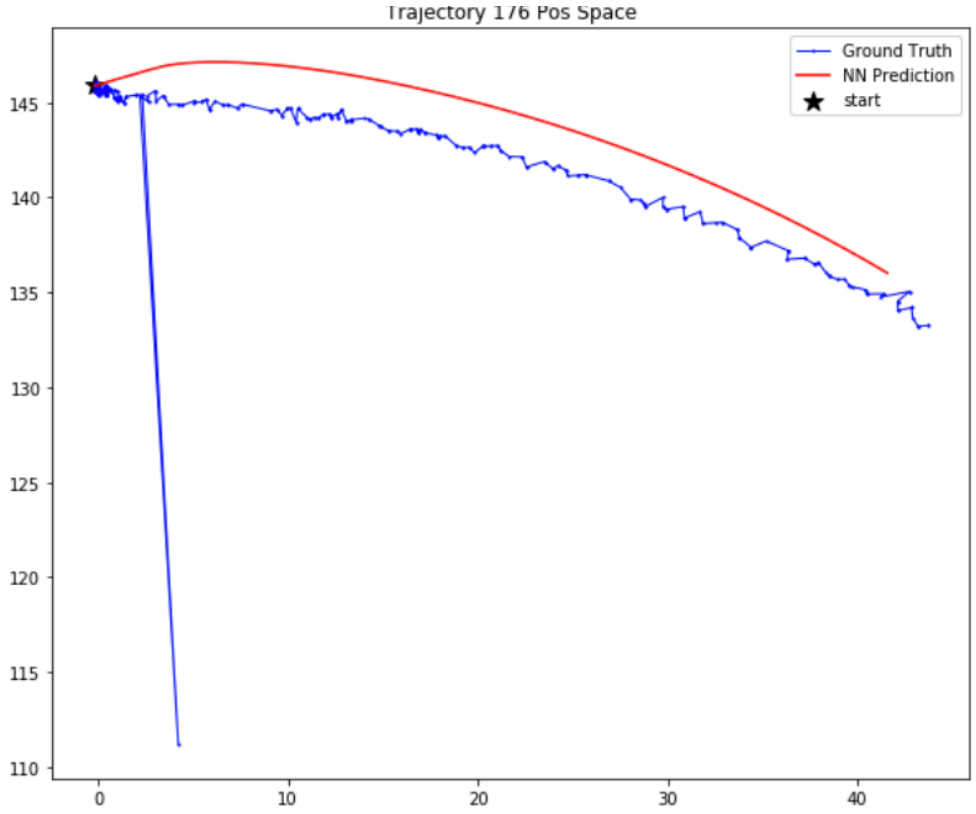
Exclude Bad Data

a) data of bad detection(tag not detected, coordinate system not parallel, …) and drop

b) data step in which position transition exceeds 1.2mm (outlier)?

Should be excluded from Training Data.

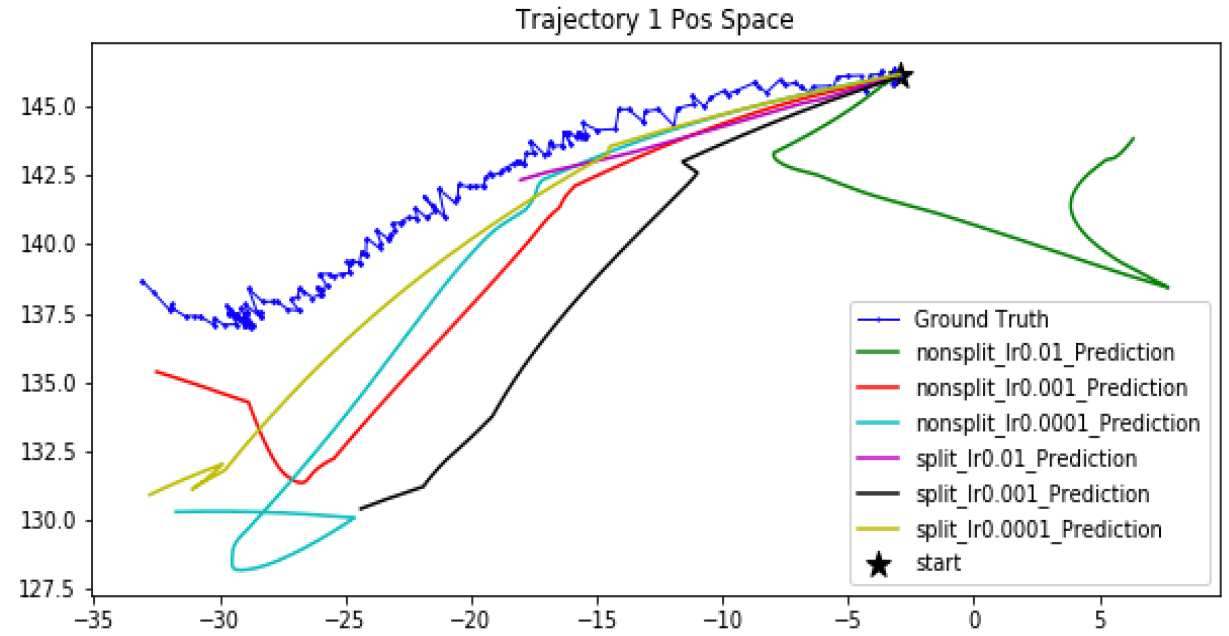


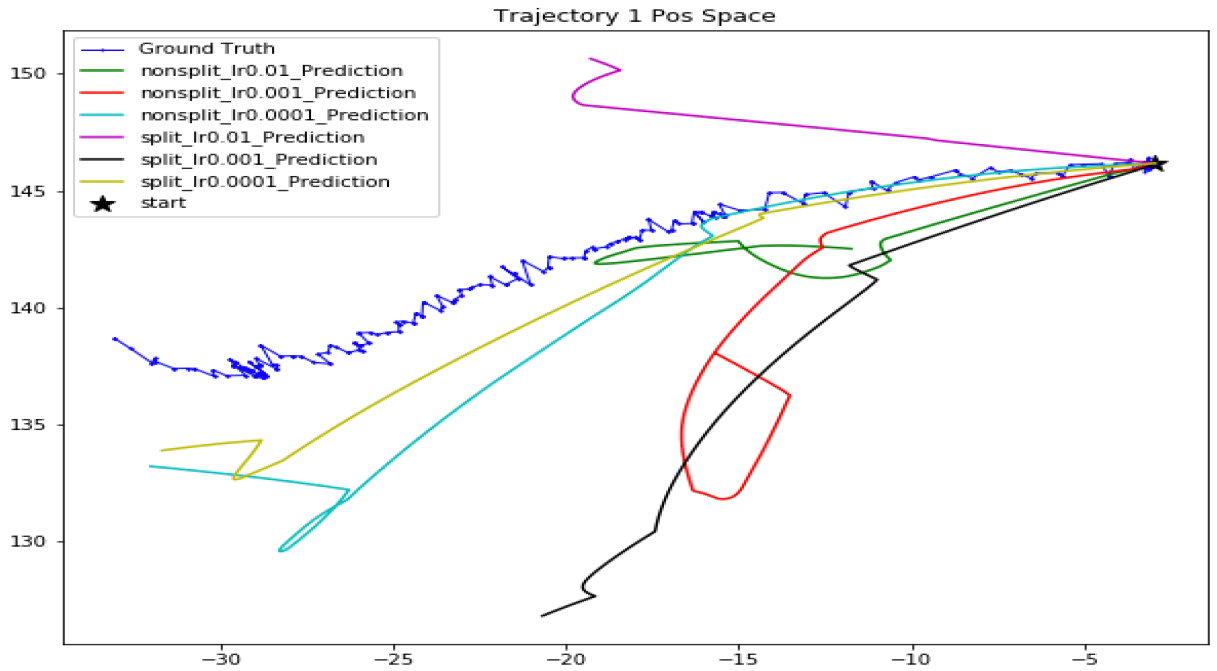


Remove such Outliers and retrain;

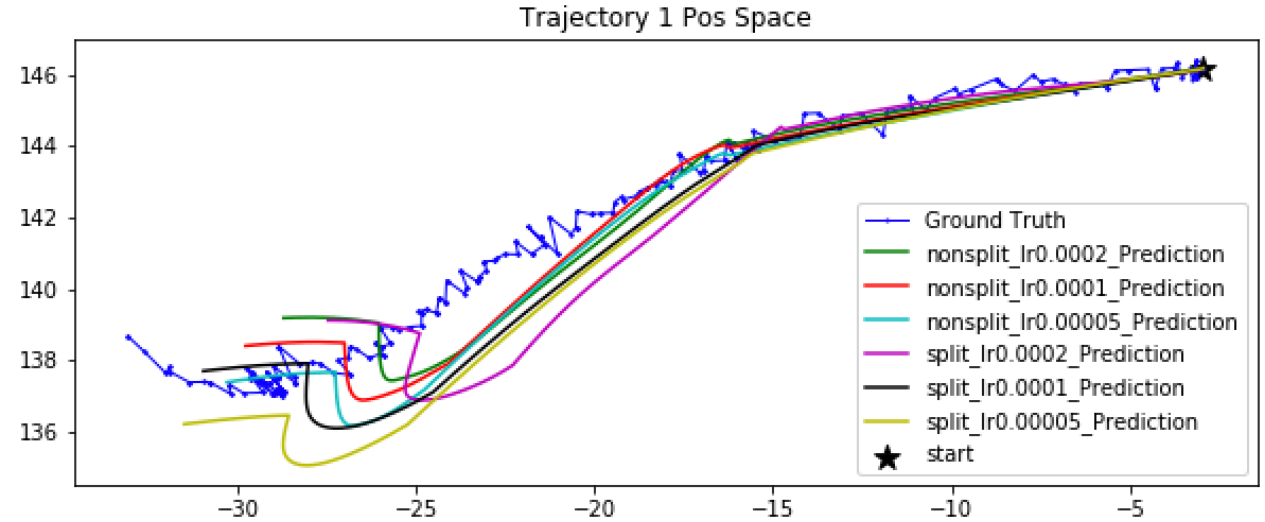
Training results until Last Week:

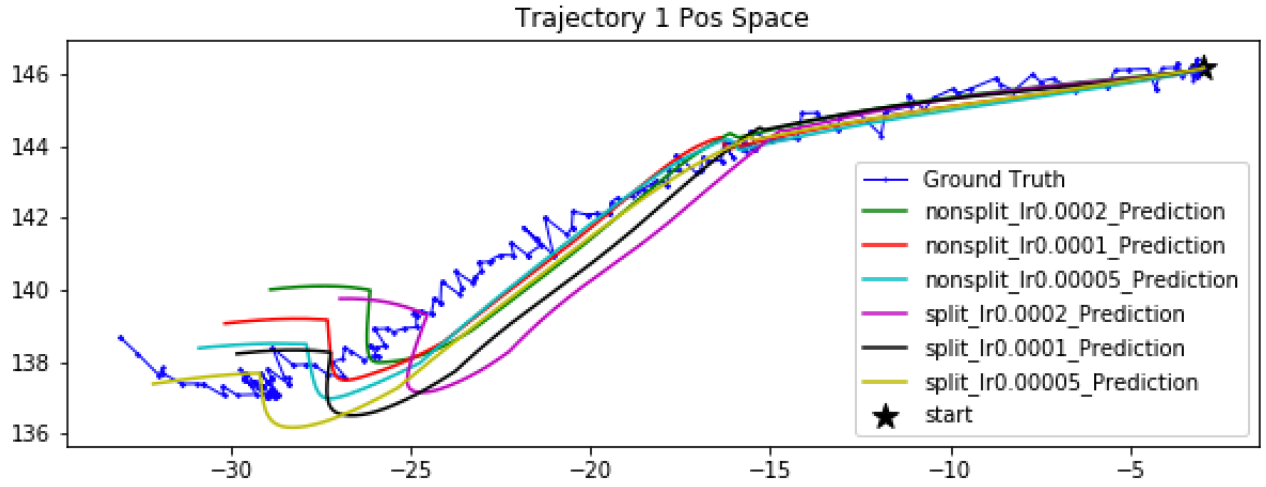
learning rate: [0.0002, 0.0001, 0.00005]; network: [split, nonsplit]; hidden nodes: [200 ,512]





Before Filter Outlier

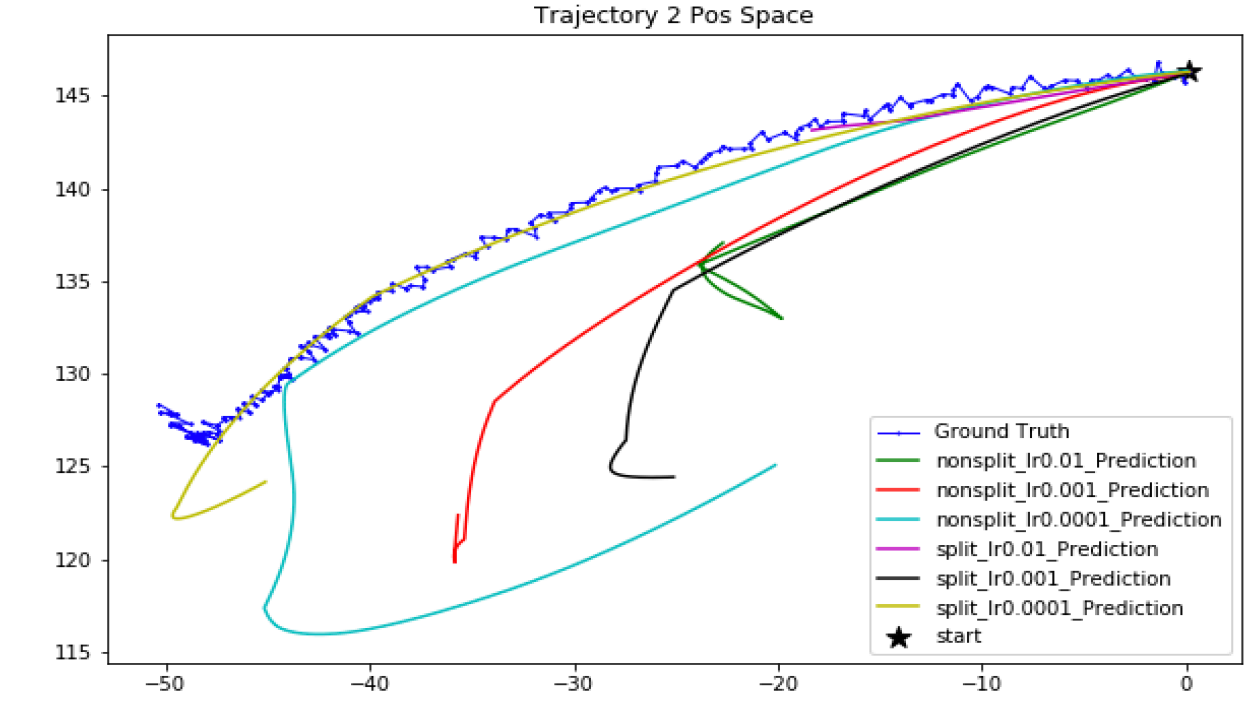


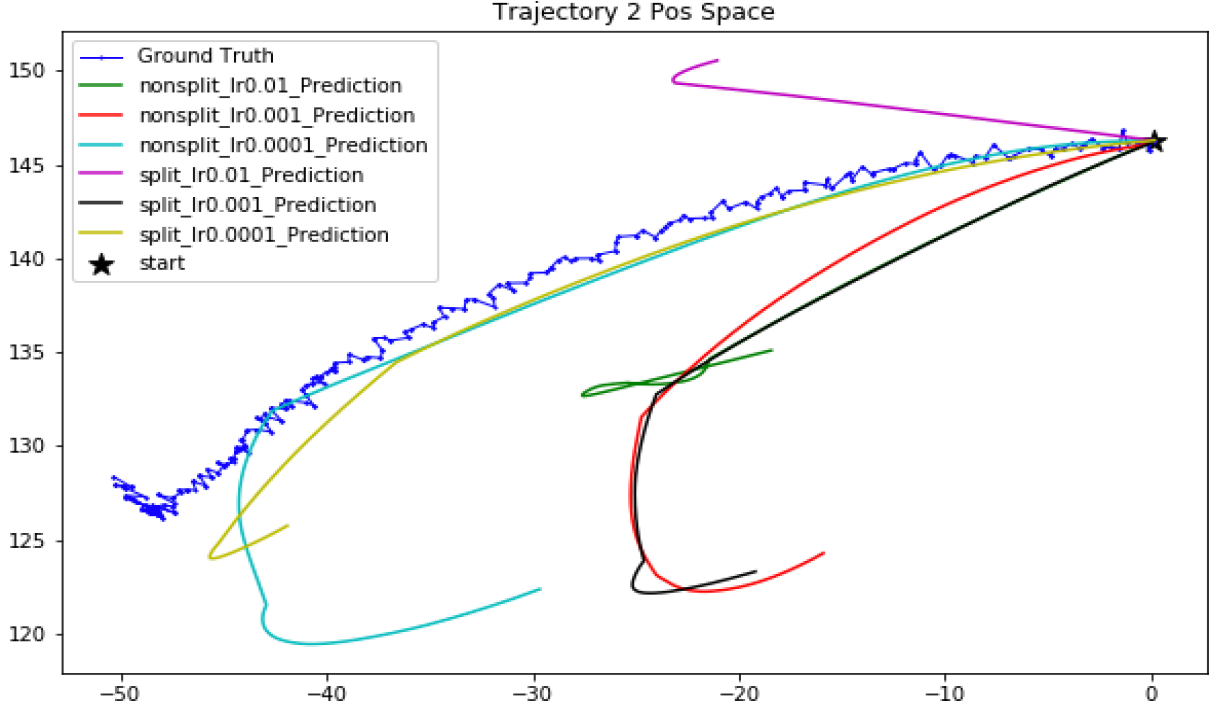


Nodes: 200

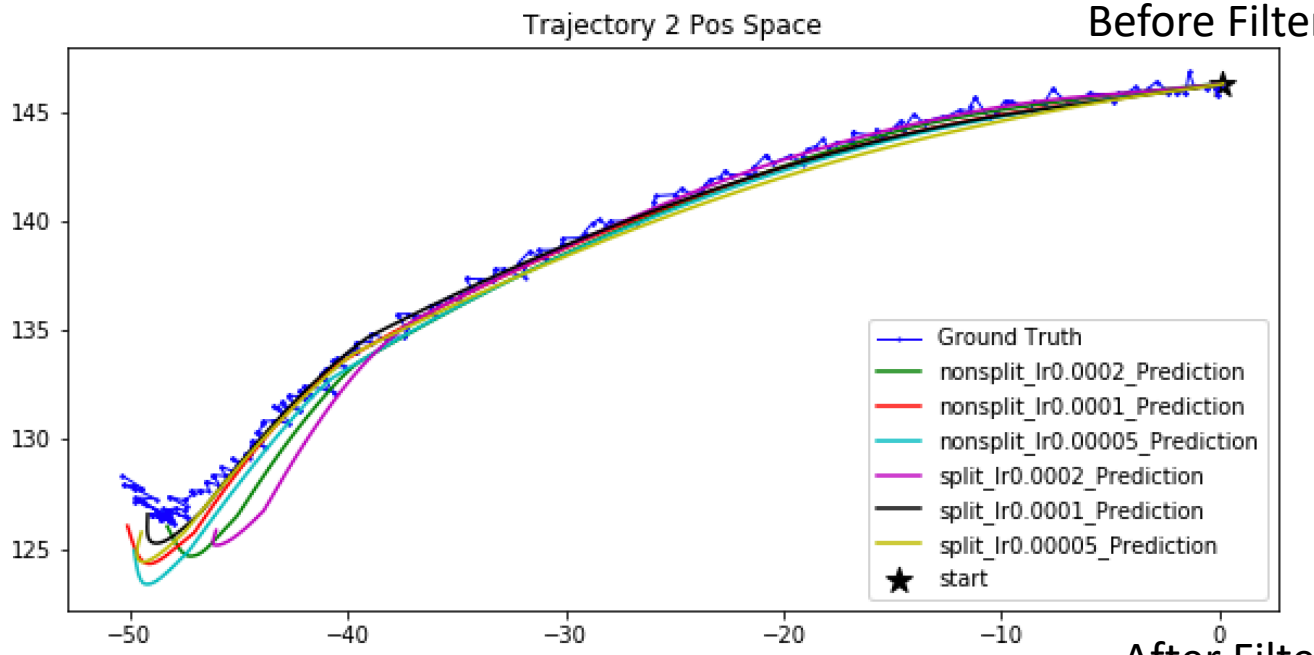
After Filter Outlier

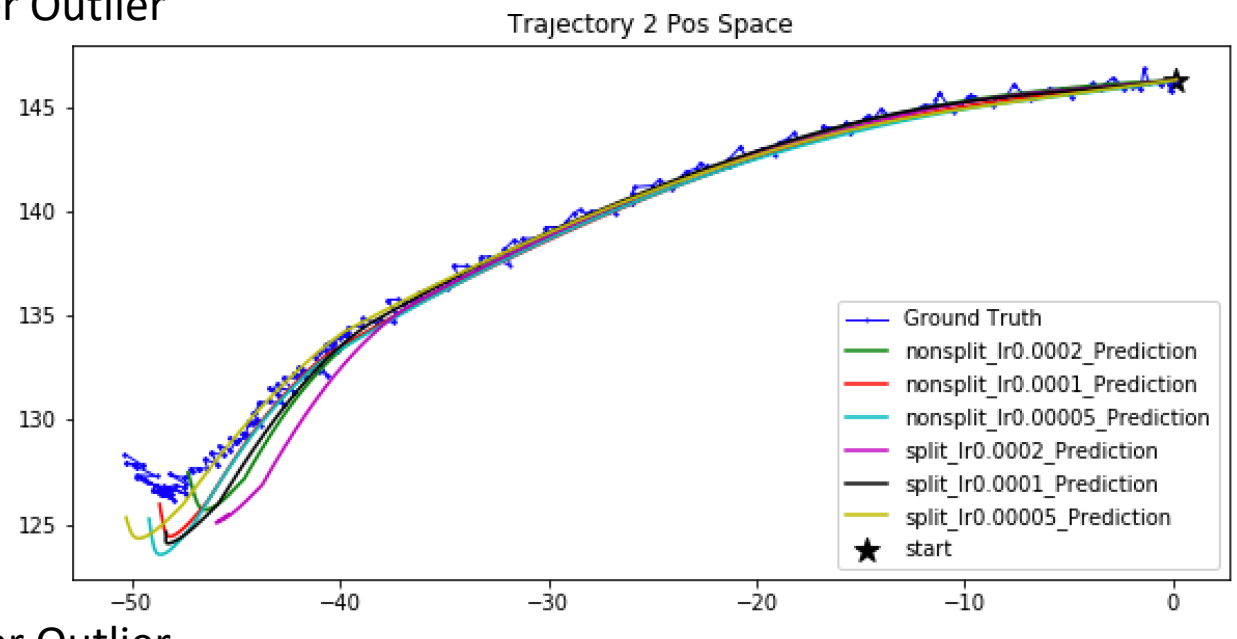
Nodes: 512





Before Filter Outlier





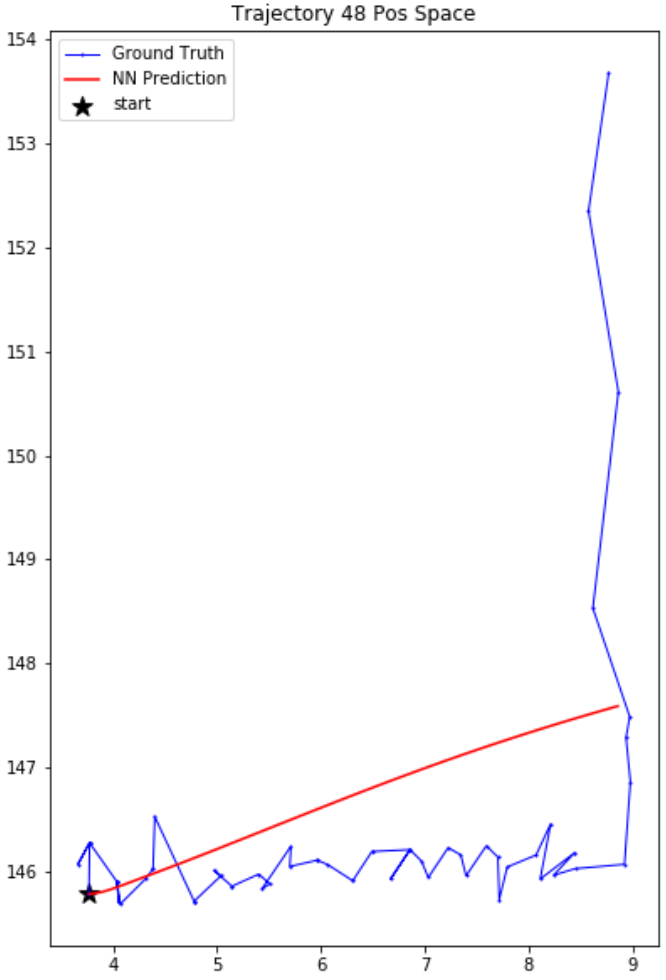
After Filter Outlier

Nodes: 200

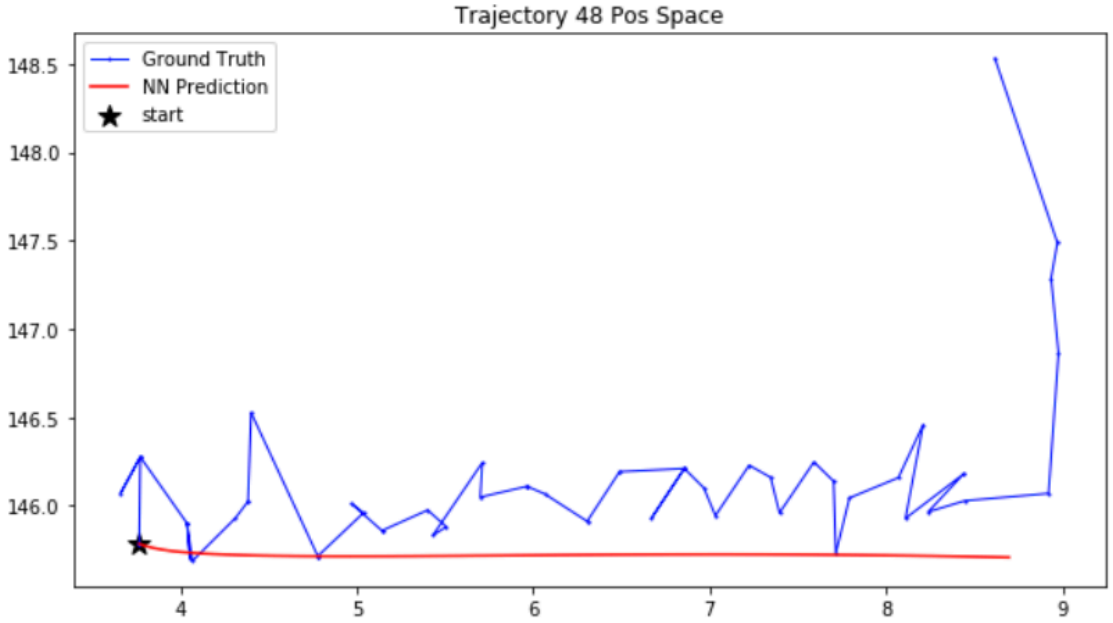
Nodes: 512

In most cases, predictions are better

Some train trajectories prediction



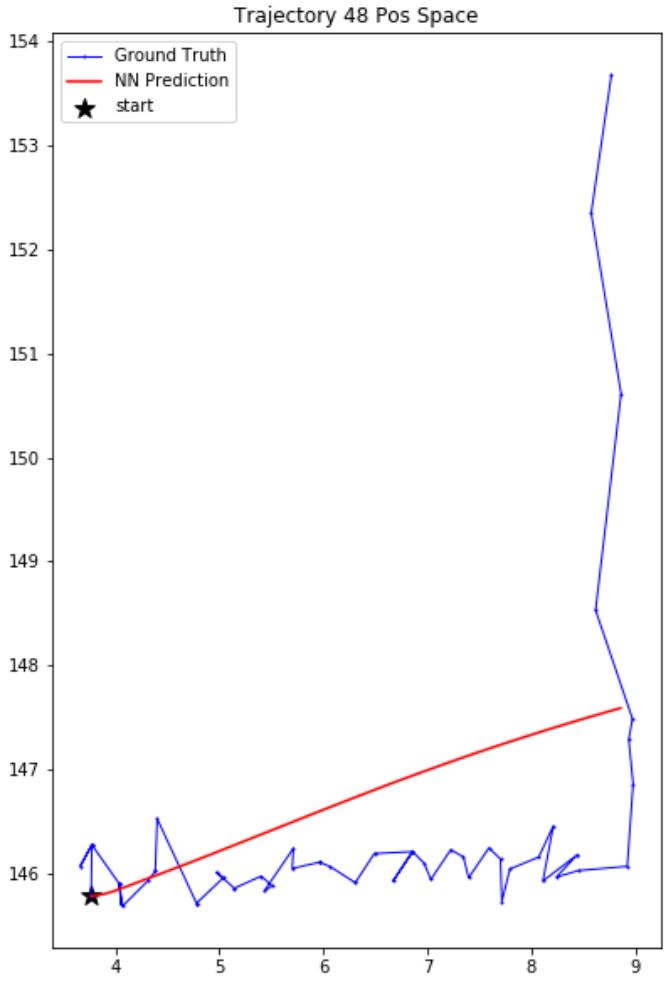
Before Filter Outlier(200 Nodes, Learning rate 0.0001, 50 Epochs)



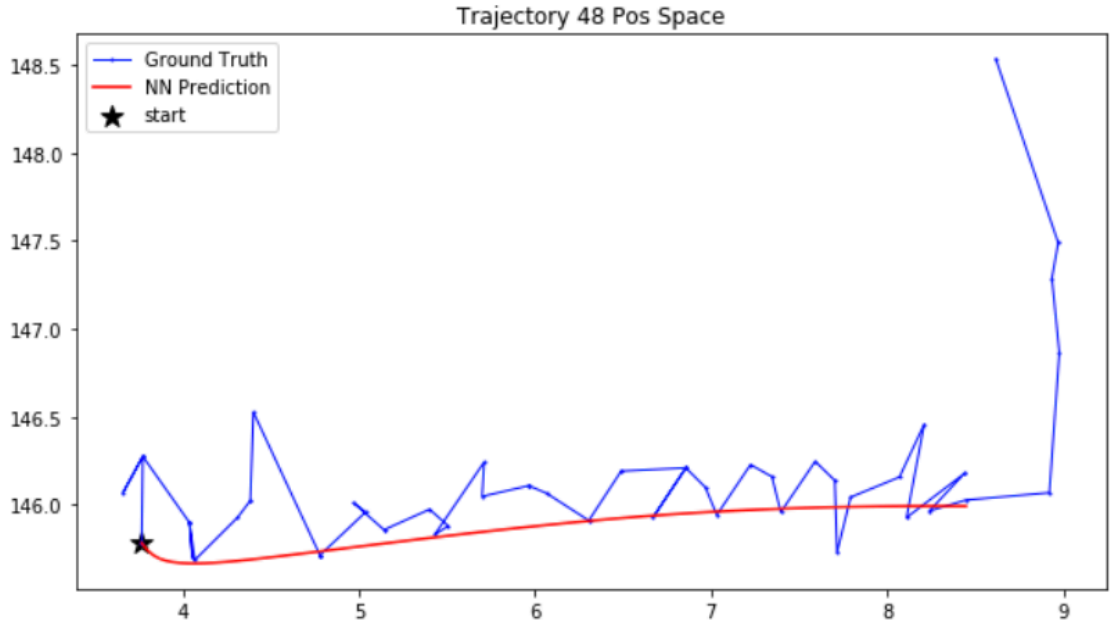
After Filter Outlier (200 Nodes, Learning rate 0.0001, 50 Epochs)

In most cases, predictions are better

Some train trajectories prediction



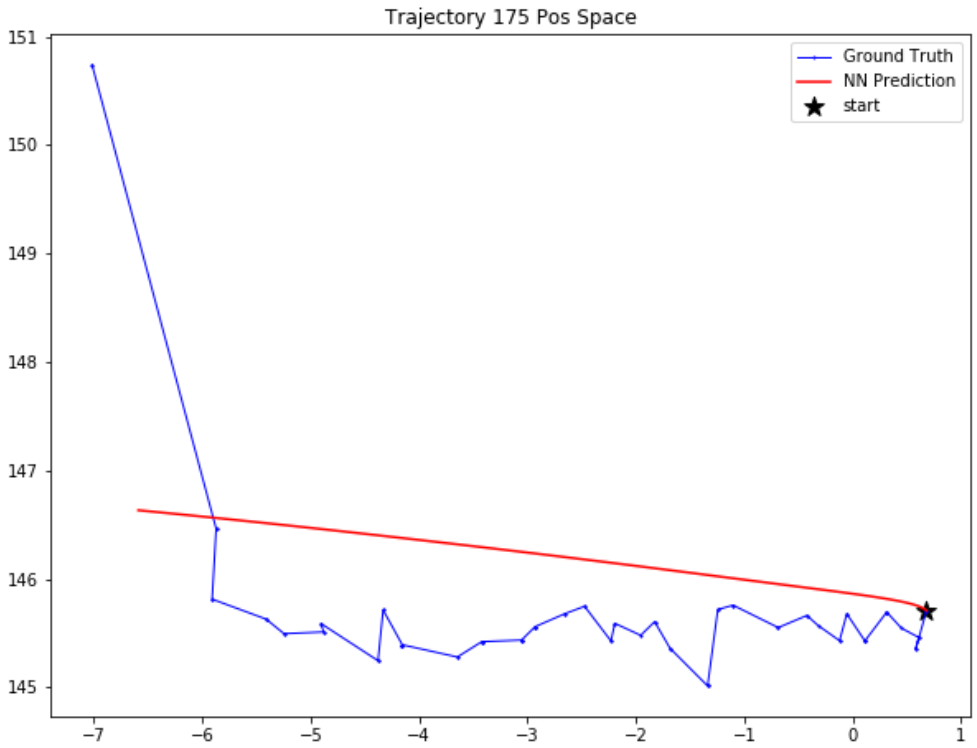
Before Filter Outlier(200 Nodes, Learning rate 0.0001,50 Epochs)



After Filter Outlier (200 Nodes, Learning rate 0.0002, 100 Epochs)

In most cases, predictions are better

Some train trajectories prediction

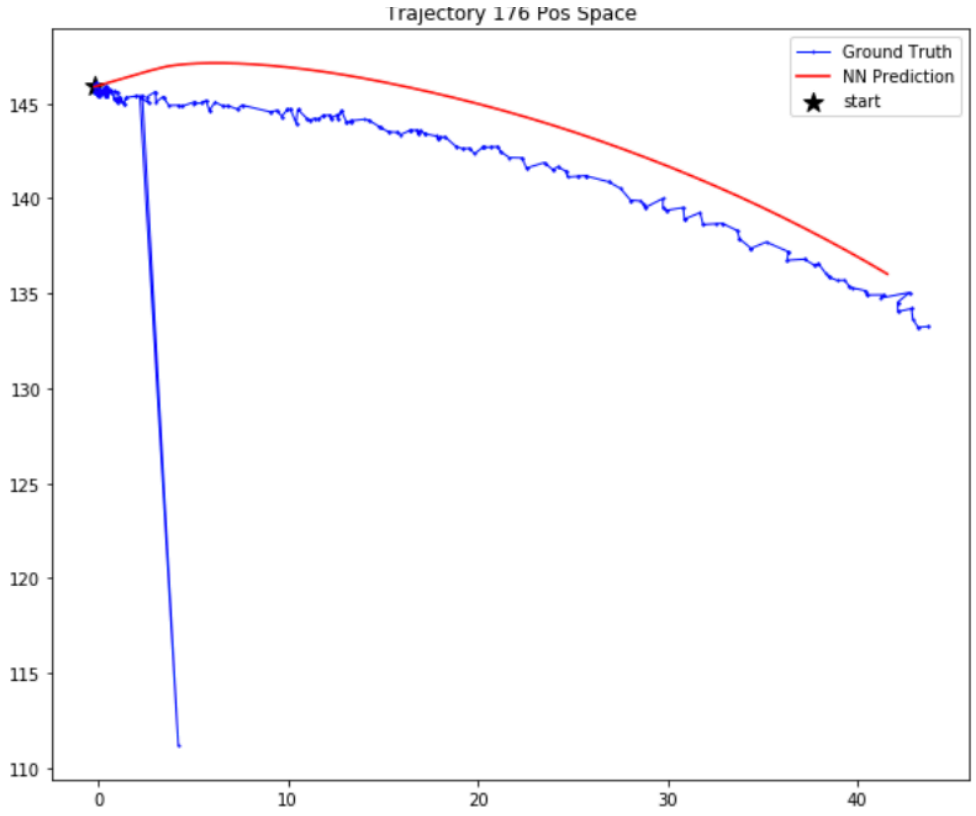


Before Filter Outlier(200 Nodes, Learning rate 0.0001, 50 Epochs)

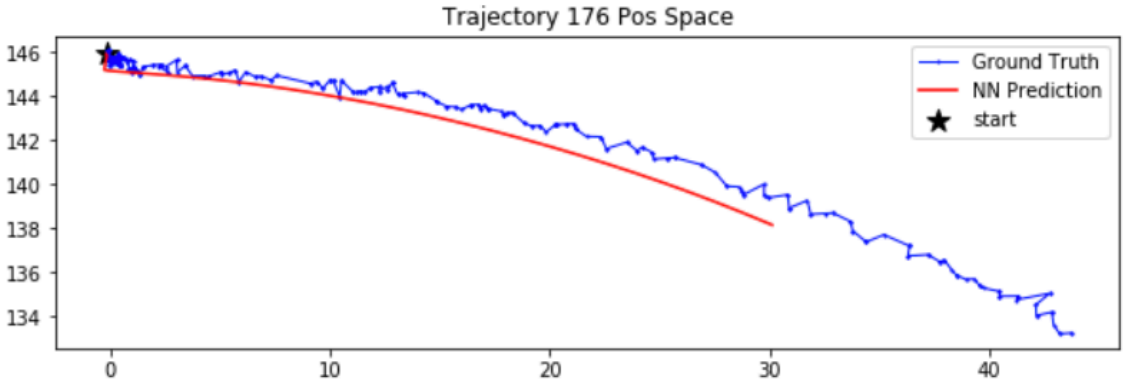
After Filter Outlier (200 Nodes, Learning rate 0.0001, 50 Epochs)

In most cases, predictions are better

Some train trajectories prediction



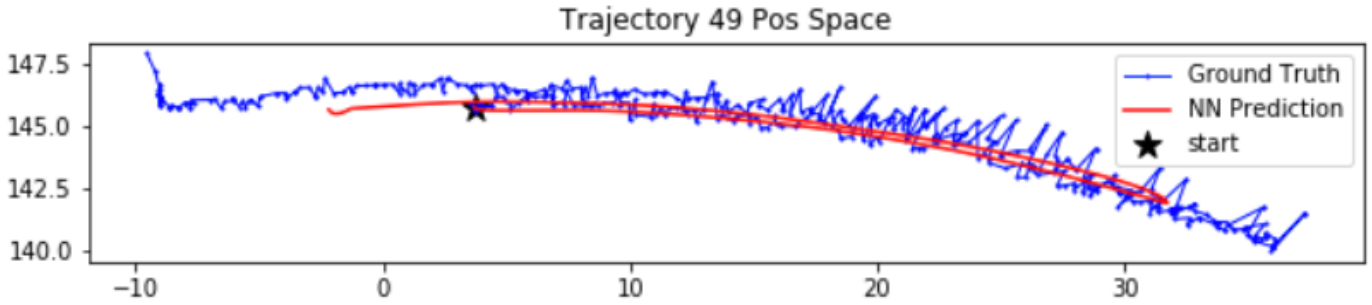
Before Filter Outlier(200 Nodes, Learning rate 0.0001, 50 Epochs)



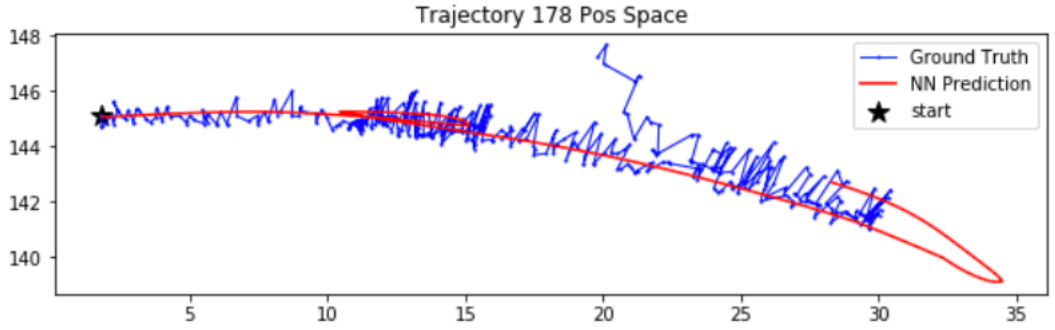
After Filter Outlier (200 Nodes, Learning rate 0.0001, 50 Epochs)

In most cases, predictions are better

Some train trajectories prediction

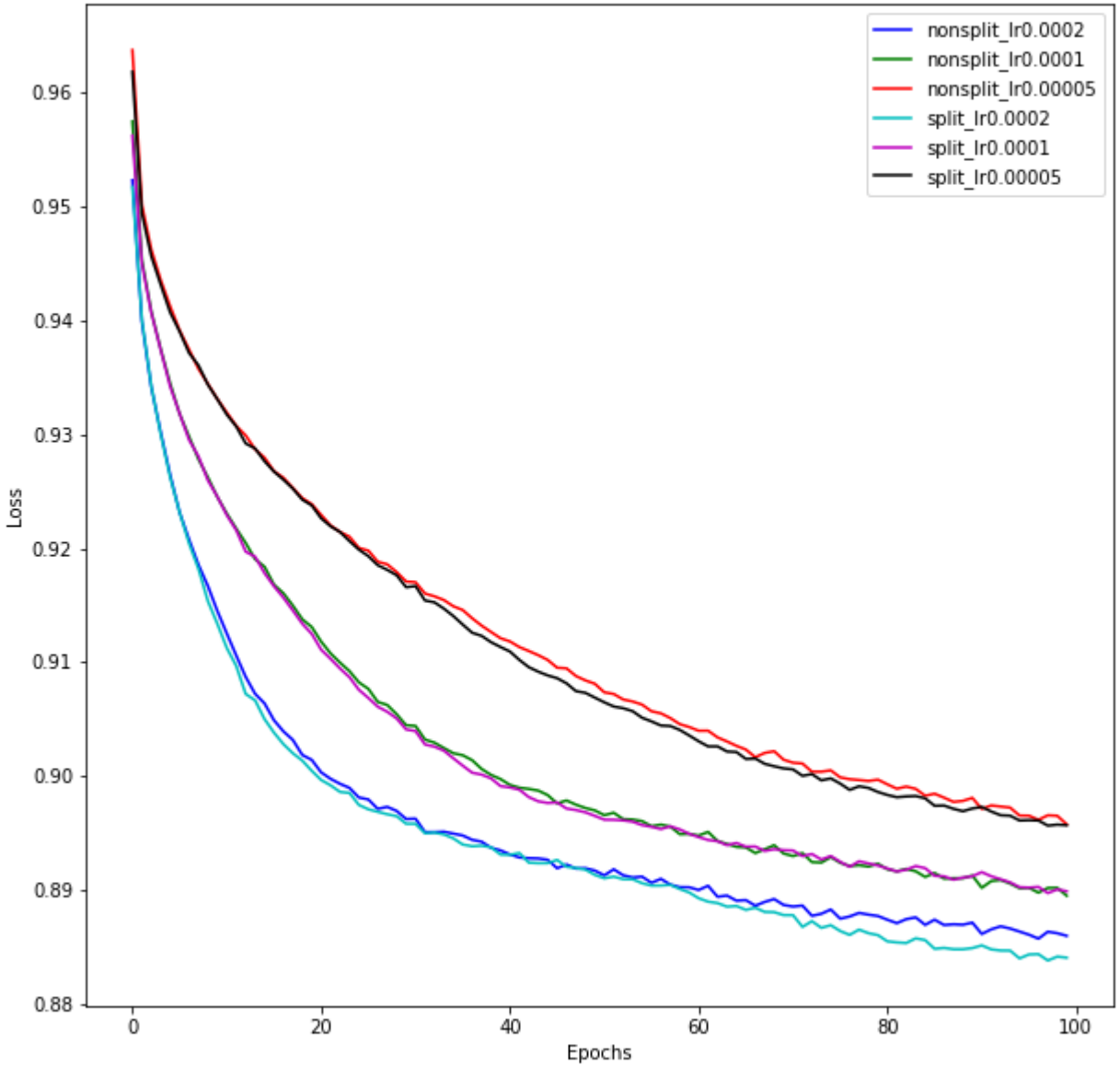


After Filter Outlier(200 Nodes,
Learning rate 0.0002,100 Epochs)

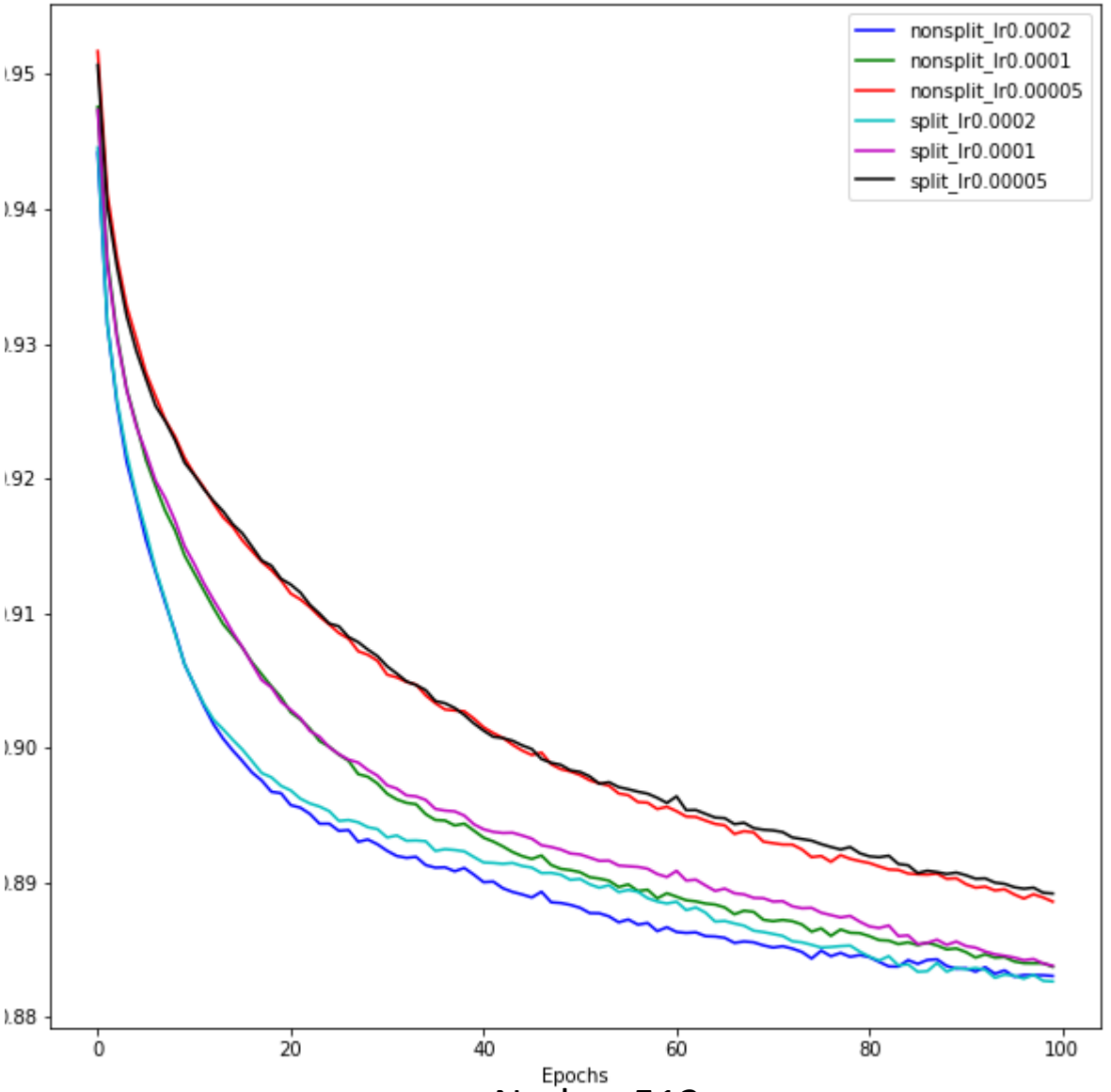


After Filter Outlier (200 Nodes,
Learning rate 0.0002, 100 Epochs)

Training Error



Nodes: 200



Nodes: 512

Exclude Bad Data

a) data of bad detection(tag not detected, coordinate system not parallel, …) and drop
b) data step in which position transition exceeds 1.2mm (outlier)

After Checked each training trajectory,

c) data step in which the previous state and the next state are exactly the same
d) data step of drastic transition at the end phase of episode (final 10 steps)
e) data from very short episodes (less than 100 steps)

Exclude Bad Data

a) data of bad detection(tag not detected, coordinate system not parallel, …) and drop
b) data step in which position transition exceeds 1.2mm (outlier)
c) data step in which the previous state and the next state are exactly the same

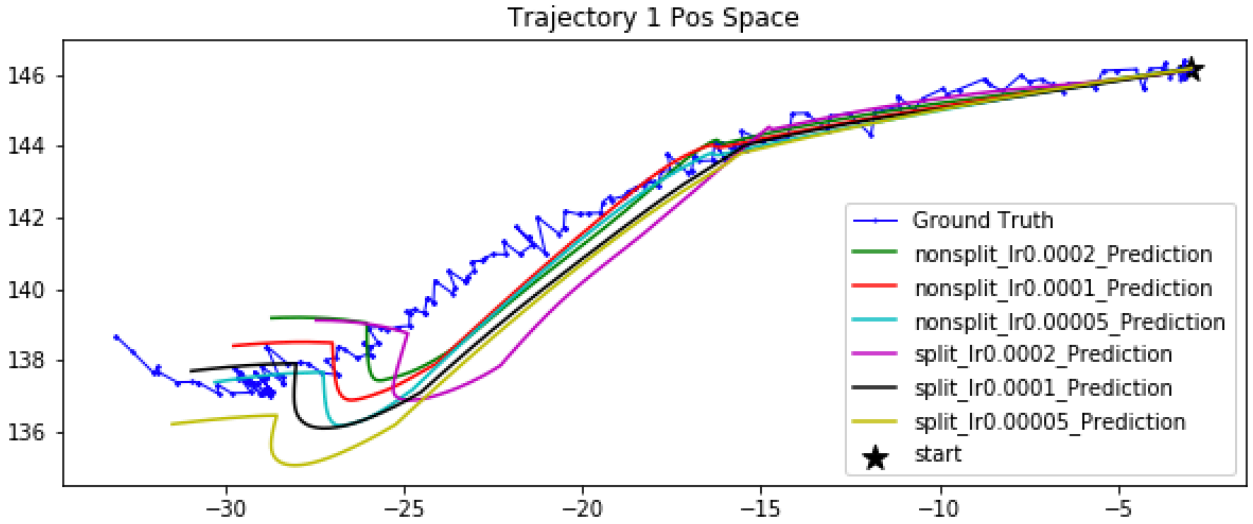
E.g. **x1 y1 LL1 RL1 → LA1 RA2 → x1 y1 LL1 RL1,**
x1 y1 LL1 RL1 → LA1 RA2 → x2 y2 LL2 RL2,
……

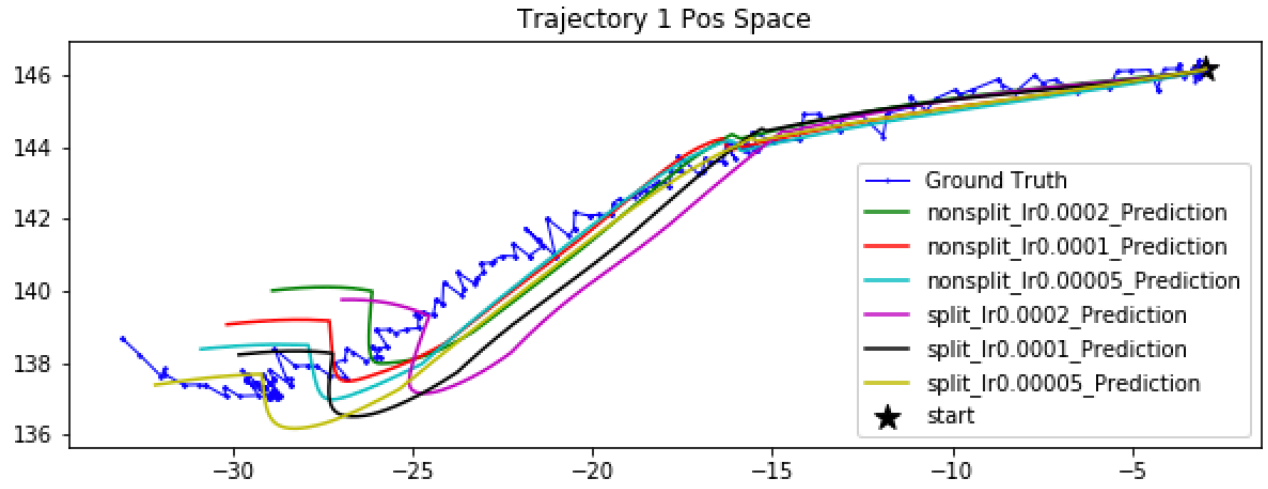
Should not happen, since if happens, one input(state) would have 2 possible outputs (next states).

They should be excluded from training data or use smoothing(mean_filter) to deal with them.

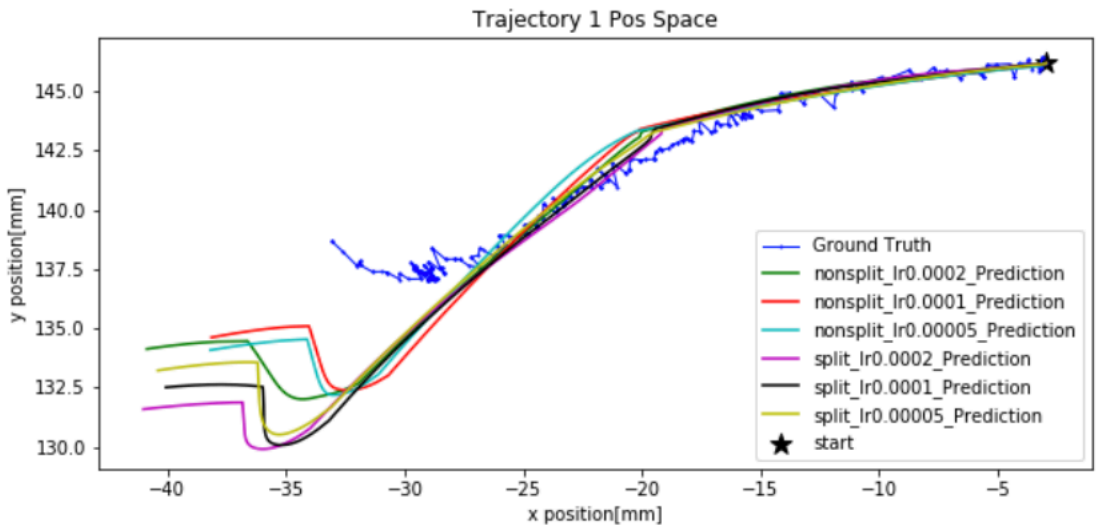
Training results after processing c) until Today:
No smoothing at all. Excluded from training data.

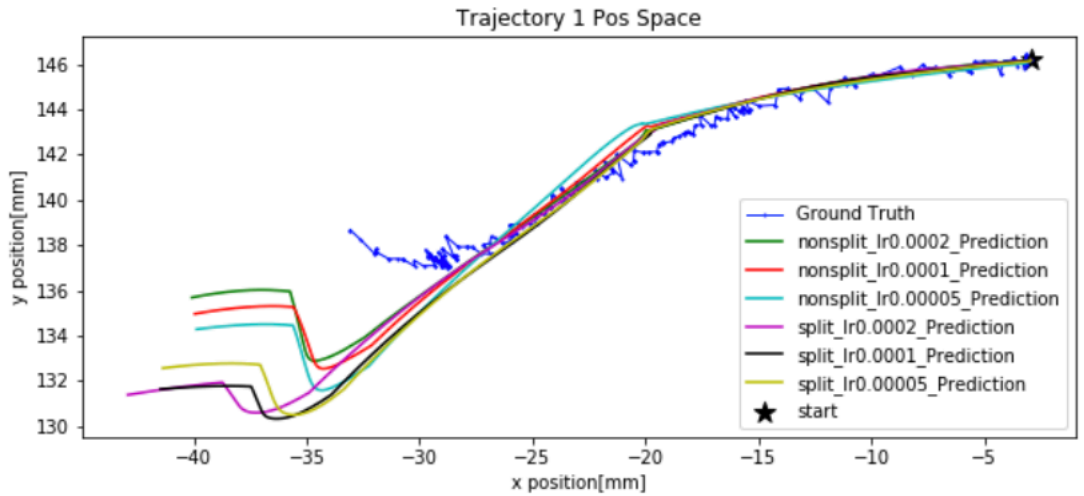
learning rate: [0.0002, 0.0001, 0.00005]; network: [split, nonsplit]; hidden nodes: [200 ,512]





Without Removing c)

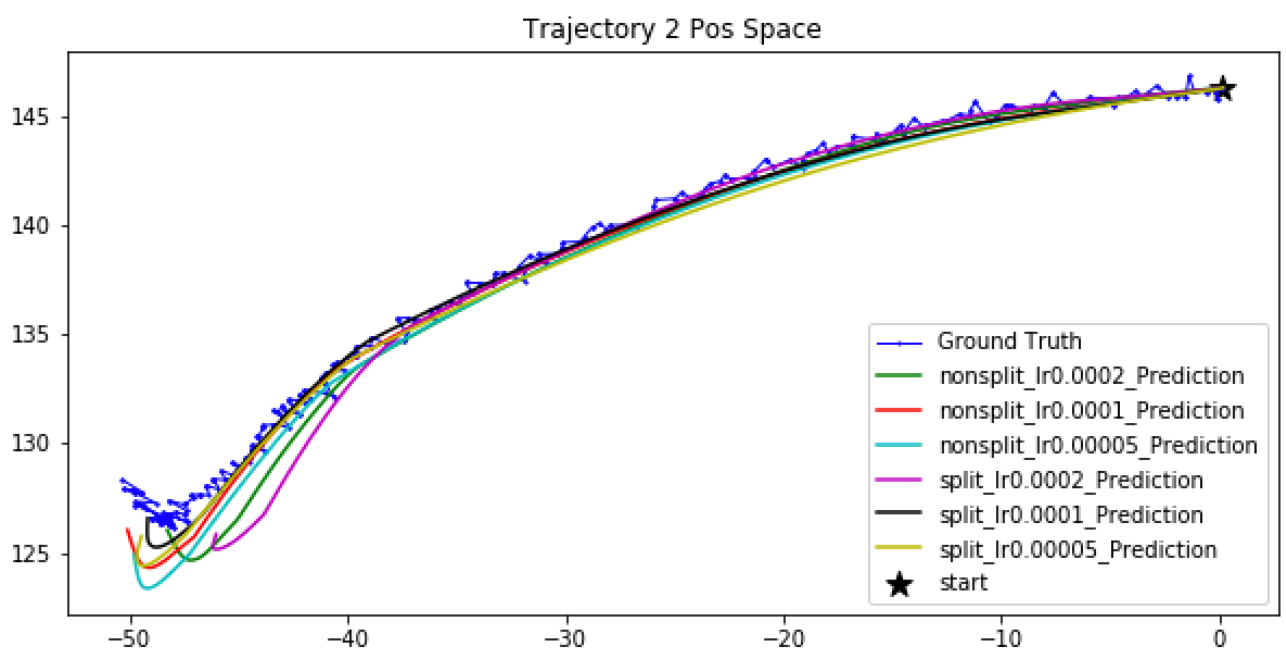


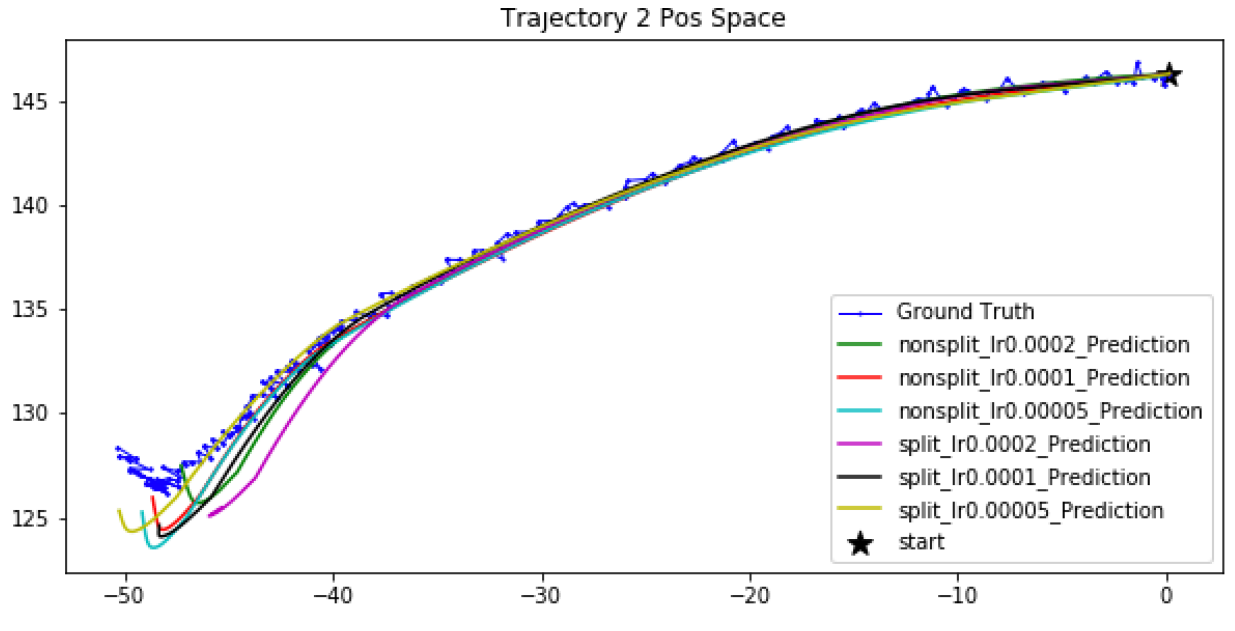


Nodes: 200

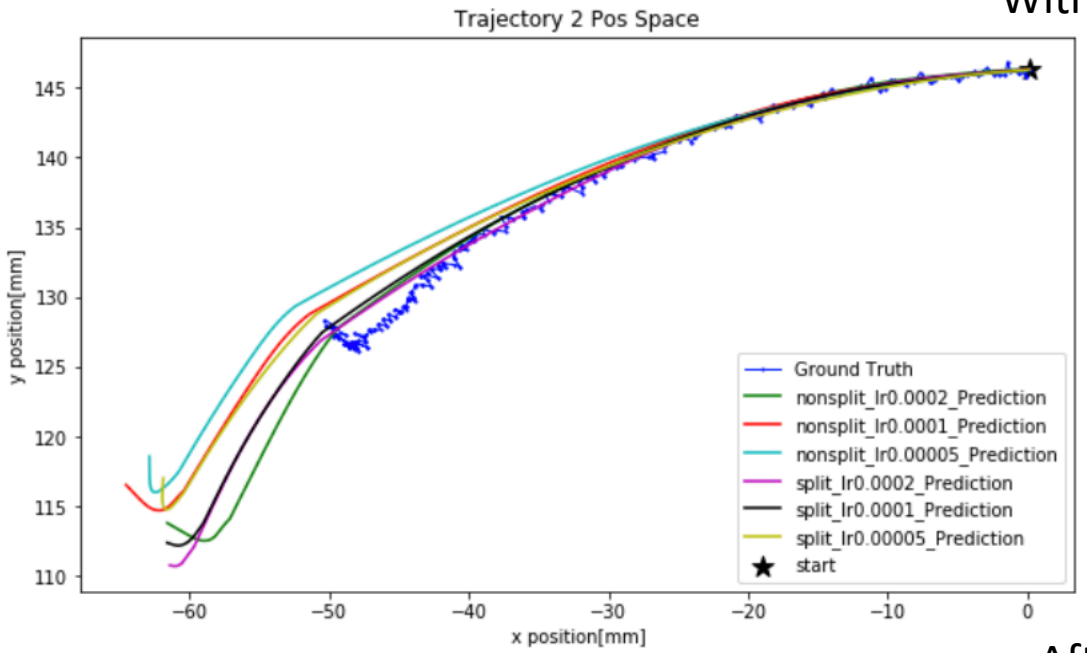
After Removing c)

Nodes: 512



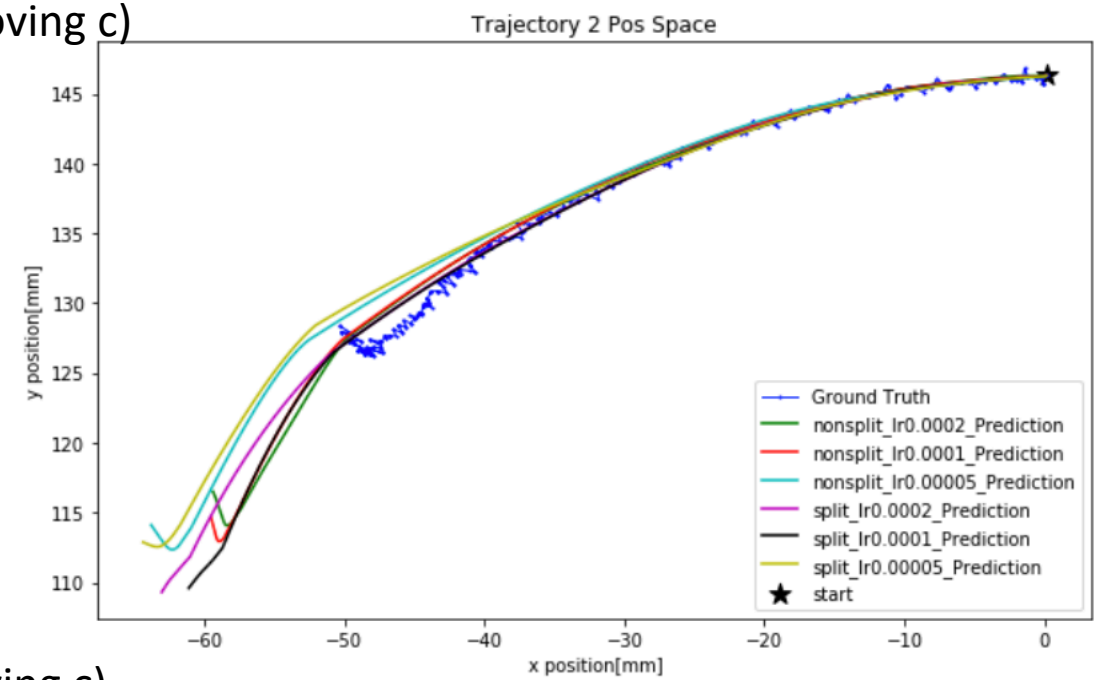


Without Removing c)



Nodes: 200

After Removing c)



Nodes: 512

Discussion

In other trajectories, predictions are also much longer than the ground truth. Possible because we have wrongly removed the state transition of data type c) which we shouldn't remove.

E.g. **x1 y1 LL1 RL1 → LA1 RA2 → x1 y1 LL1 RL1,**
x1 y1 LL1 RL1 → LA1 RA2 → x2 y2 LL2 RL2,
……

In the example, two steps of action pairs (**LA1 RA2**) would transit **x1 y1 LL1 RL1** to **x2 y2 LL2 RL2** . However, if we delete the first step, we falsely assume only one step of action pairs (**LA1 RA2**) would transit **x1 y1 LL1 RL1** to **x2 y2 LL2 RL2** . Thus, we got longer prediction trajectory.

To avoid this problem, **Smoothing!**

Exclude Bad Data

a) data of bad detection(tag not detected, coordinate system not parallel, …) and drop
b) data step in which position transition exceeds 1.2mm (outlier)

After Checked each training trajectory,

c) data step in which the previous state and the next state are exactly the same
d) data step of drastic transition at the end phase of episode (final 10 steps)
e) data from very short episodes (less than 100 steps)

Training is still running after processing d) and e)

learning rate: [0.0002, 0.0001, 0.00005]; network: [split, nonsplit]; hidden nodes: [200 ,512]

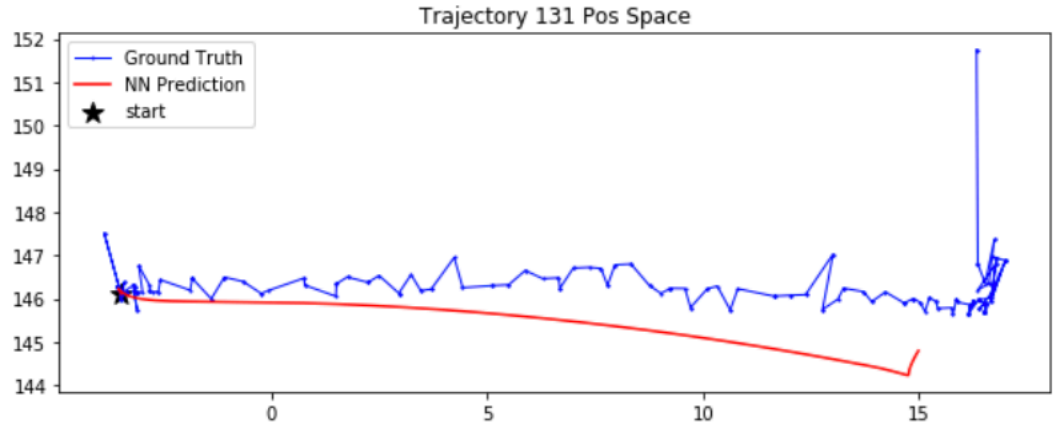
Exclude Bad Data

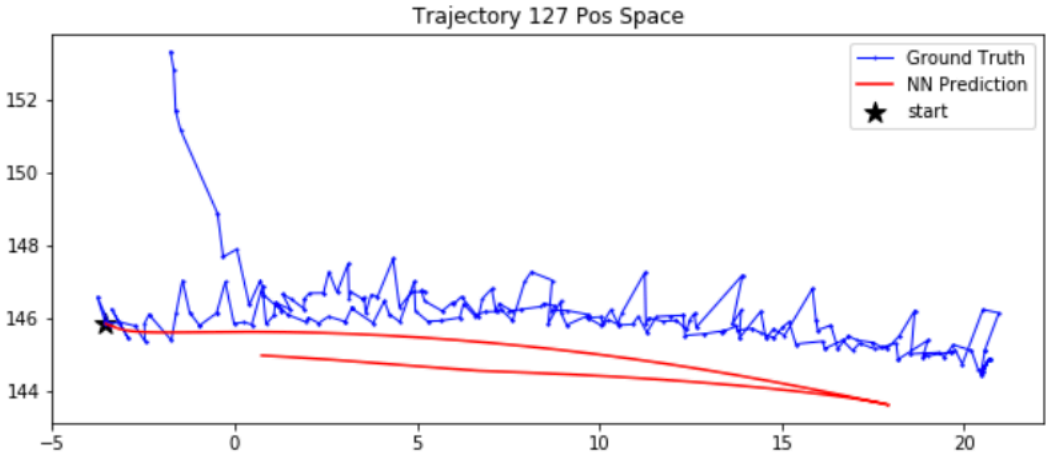
a) data of bad detection(tag not detected, coordinate system not parallel, …) and drop
b) data step in which position transition exceeds 1.2mm (outlier)
c) data step in which the previous state and the next state are exactly the same
d) data step of drastic transition at the end phase of episode (final 10 steps)

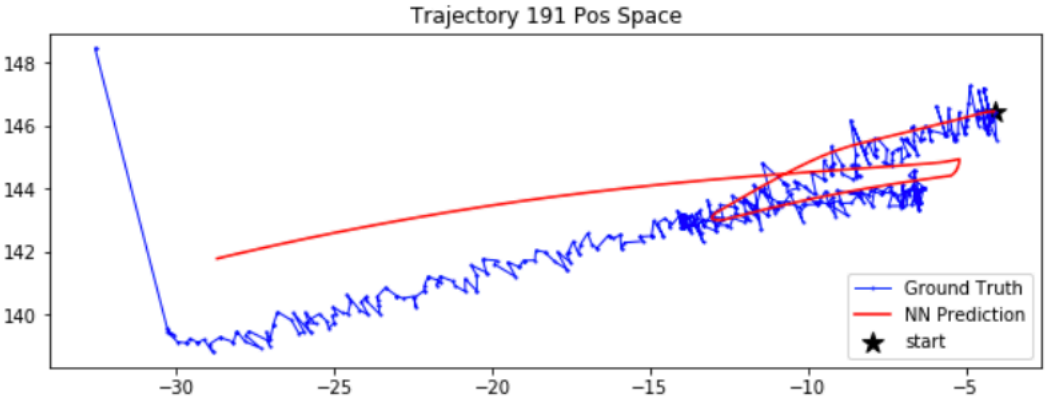
Happened, possibly due to unstable transition and visual detection at the end of an episode, though the visual detection worked well and did not detect drop.

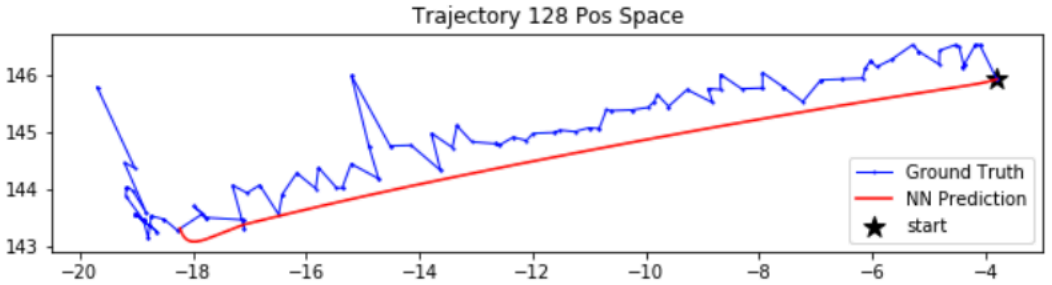
Discussion:
Should be excluded from training data?
Current training: excluded

Examples (Bad Tails)









Nonsplit Model, 200 Nodes, Learning rate 0.0002

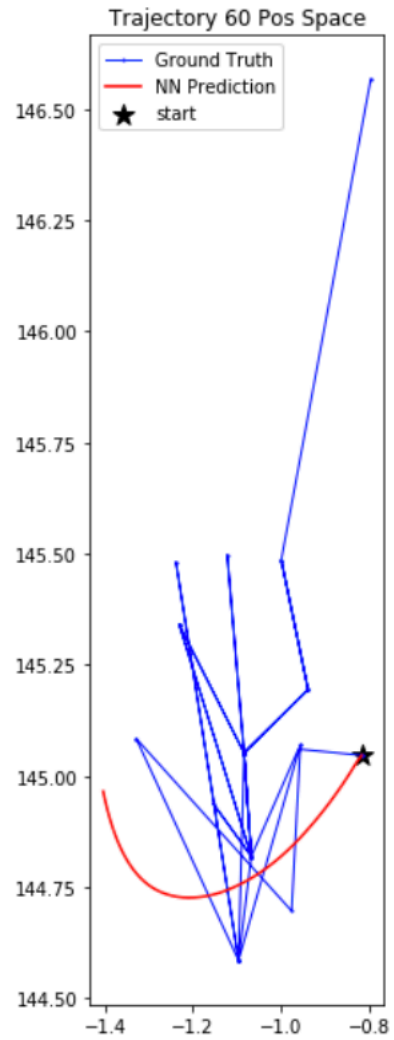
Exclude Bad Data

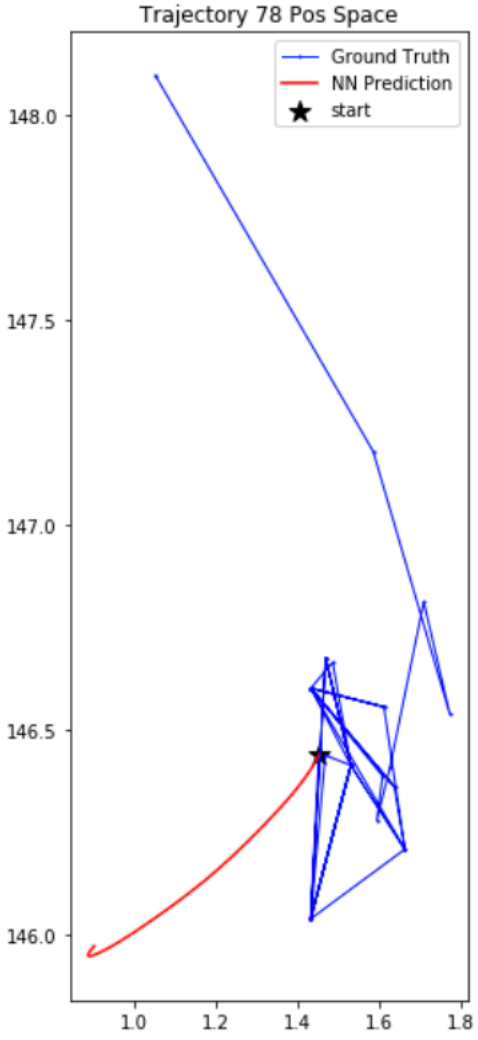
a) data of bad detection(tag not detected, coordinate system not parallel, …) and drop
b) data step in which position transition exceeds 1.2mm (outlier)
c) data step in which the previous state and the next state are exactly the same
d) data step of drastic transition at the end phase of episode (final 10 steps)
e) data of very short episodes (less than 100 steps)

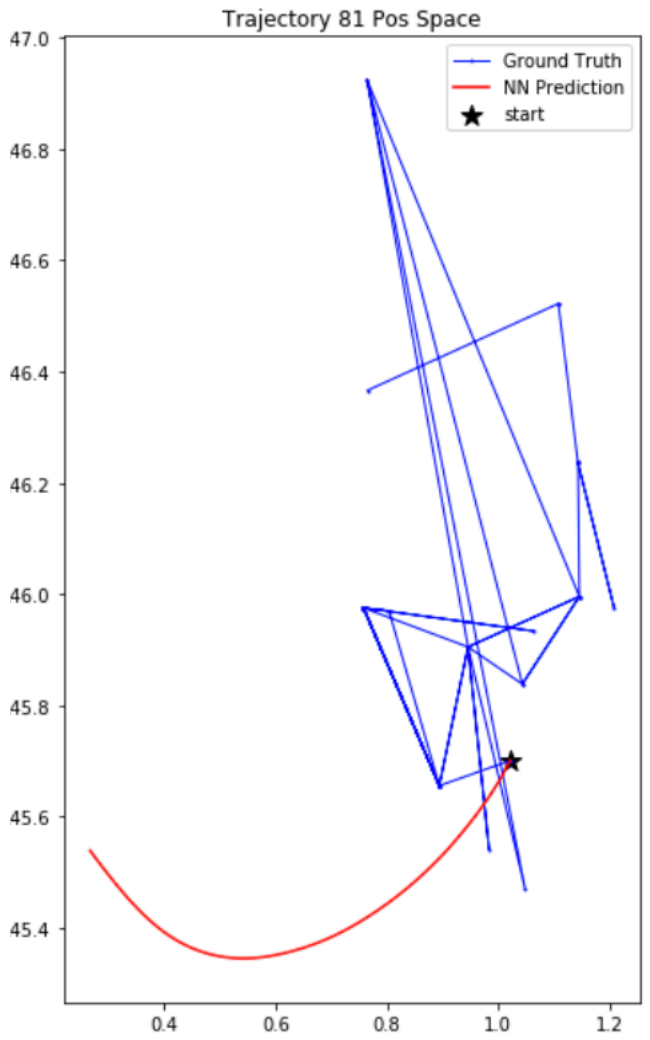
A very short episode could appear if the general visual detection in that episode is very bad and thus only few valid data steps are collected. Finally, they became very messy.

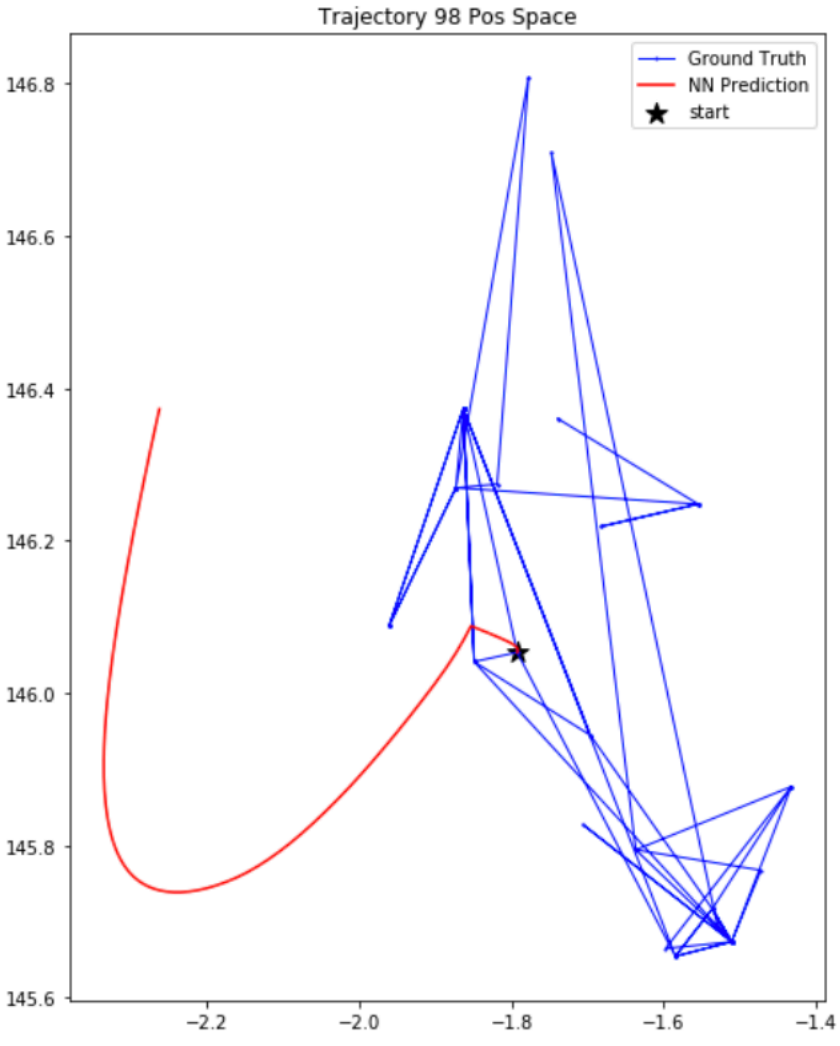
Discussion:
Should be excluded from training data?
Current training: excluded

Examples (Short Episodes, Very Messy)









Nonsplit Model, 200 Nodes, Learning rate 0.0002

If Smoothing

Types of Bad Training Data

a) data of bad detection and drop?

b) data step in which position transition exceeds 1.2mm (outlier)?

c) data step in which the previous state and the next state are exactly the same?

d) data step of drastic transition at the end phase of episode (final 10 steps)?

e) data of very short episodes (less than 100 steps)?

Discussion:

1) Before Smoothing, also remove data of type b) and d) besides a) and then recalibrate?

2) When Smoothing, should skip data of type a), b) and d).
(e.g. smoothing window of 40, only 20 are valid, then take average of valid 20 data)
Also smooth the start state?

3) Smoothing could eliminate data of type c).
After Smoothing, remove data of type e)?

Next Plans

0) **Wait training results after processing data type of d) and e)**

1) **Data Smoothing (Recalibration?+Processing d) and e)?) + Retrain**

2) **Read papers carefully + think about further exploration of the hand project**