# Deep Learning for Autonomous Driving Spring 2021

## Project 3: 3D Object Detection from Lidar Point Clouds

Group members: Xiao'ao Song(18-747-949) Qi Ma(20-960-225)

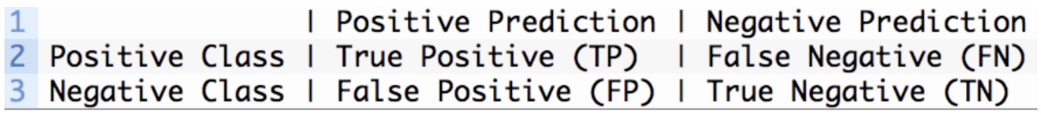
## Problem 1: Building a 2 Stage 3D Object Detector

**1.1 Given a threshold of t = 0.5, compute the average recall of the first stage network proposals for the val-set:**

a) **What is the average recall for the val-set?**
**Answer:**
The average recall for the val-set is 80.1%.

b) **Why is recall a good metric to assess the quality of the first stage proposals (as opposed to e.g. average precision)?**
**Answer:**

Here is a chart of the confusion matrix:



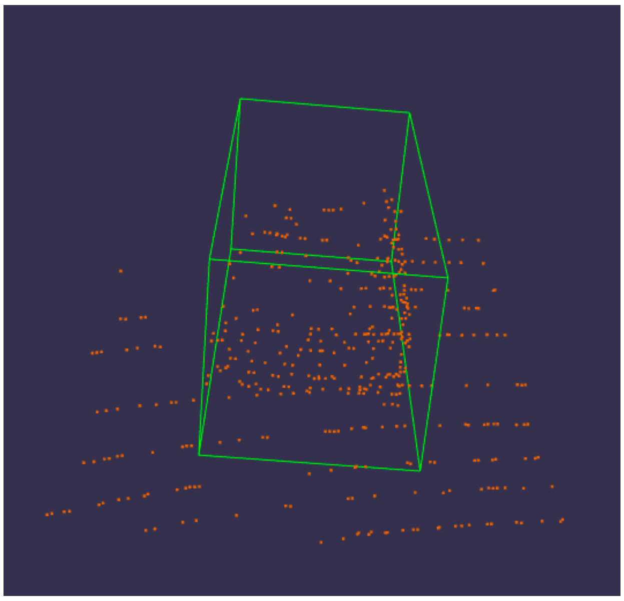
| | Positive Prediction | Negative Prediction |
|---|---|---|
| Positive Class | True Positive (TP) | False Negative (FN) |
| Negative Class | False Positive (FP) | True Negative (TN) |

By definition, recall = TP / (TP+FN) and precision = TP / (TP+FP). Most time precision and recall are often in tension. That is, improving precision typically reduces recall and vice versa.
For this trade off, we prefer a high recall over a high precision at the first stage because of two reasons: 1) it brings us more cost or even a disaster for a low recall. Imagine when there is a car but you did not perceive it, then you might crash with it. On the other hand, if there is no car and you think a car is present there and try not to go to that area, this does not really bring you any trouble. 2) In the step, the setting of the threshold is in favor of a high recall, and thus results in a lower precision. However, we still can fine tune it to get a better precision in the later stage.

## 1.2 Pool ROI's to form the input for the second stage model:

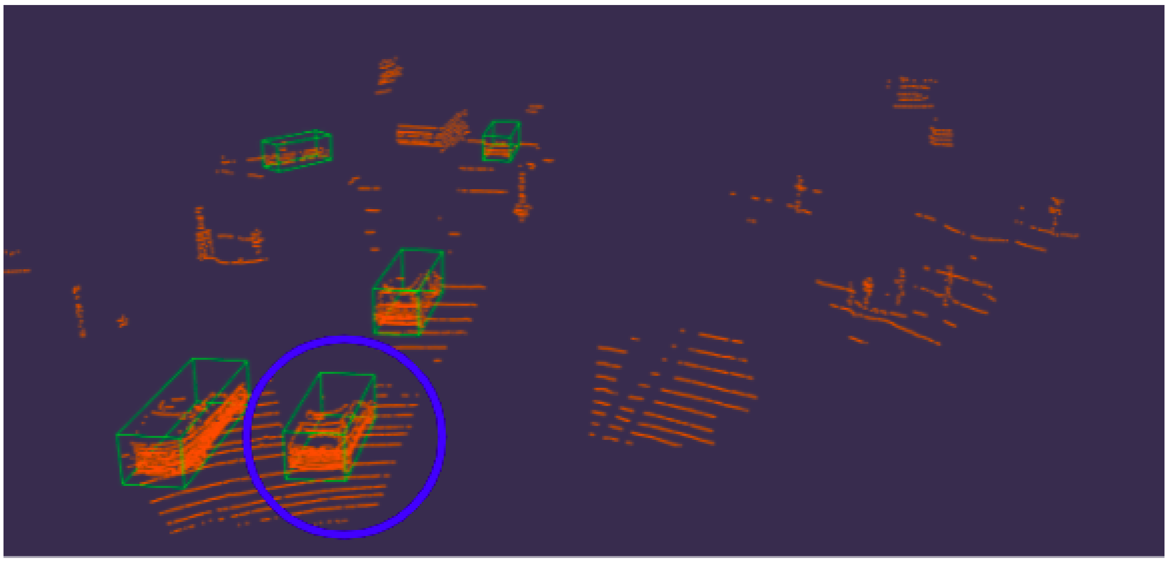
**Generate 3 such examples of ROI figures including the points and bounding box from 3 different scenes. In your report, include the frame names and the bounding box parameters corresponding to each figure**

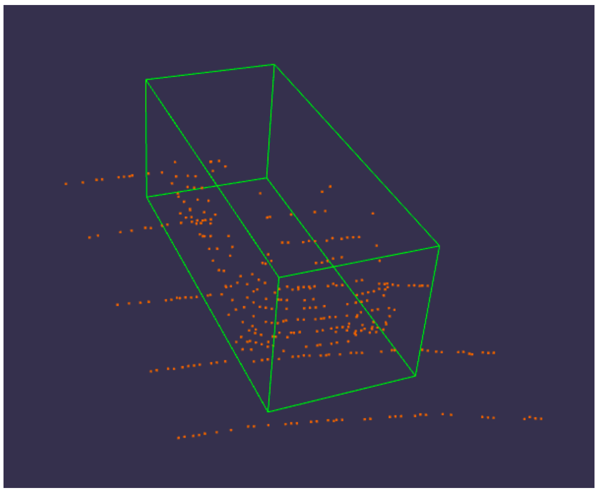
Visualization 3 scenes:



frame: elpBlgmdIiJkAAVuYPB2

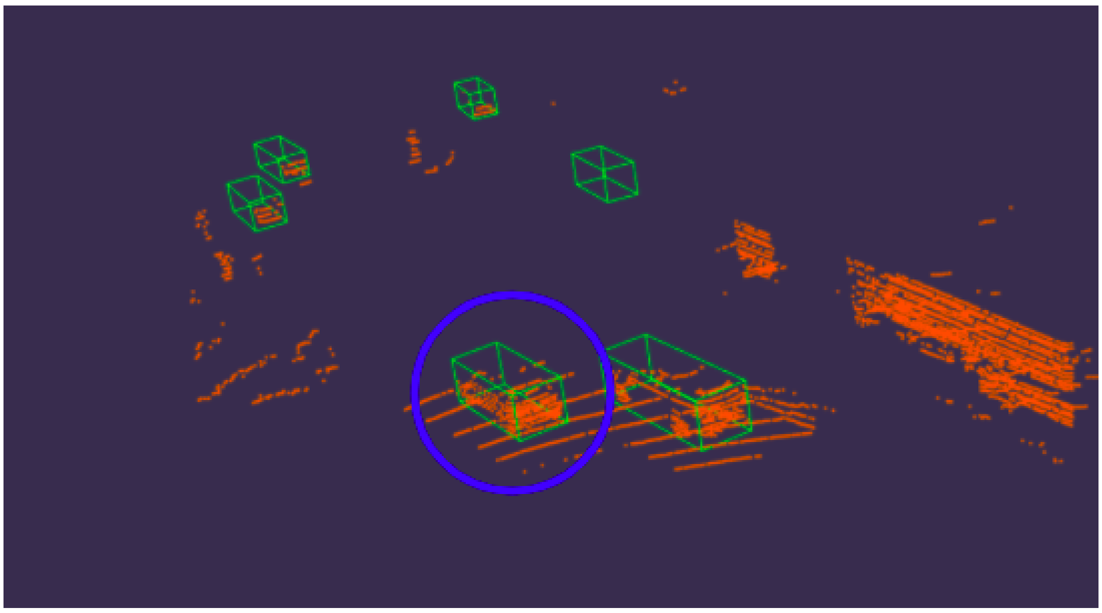
bounding box: [ 3.5176,  1.6643, 11.0931,  1.5604,  1.6181,  3.7449, -1.5691]

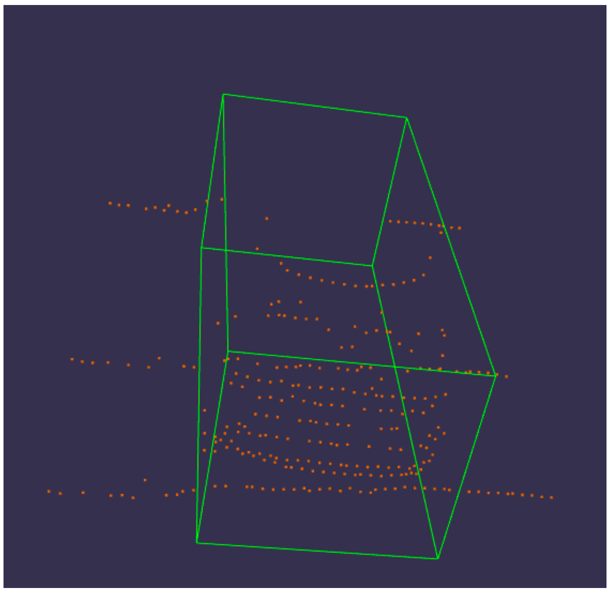




frame:  rWvdEifFU9YpFeFyMroa

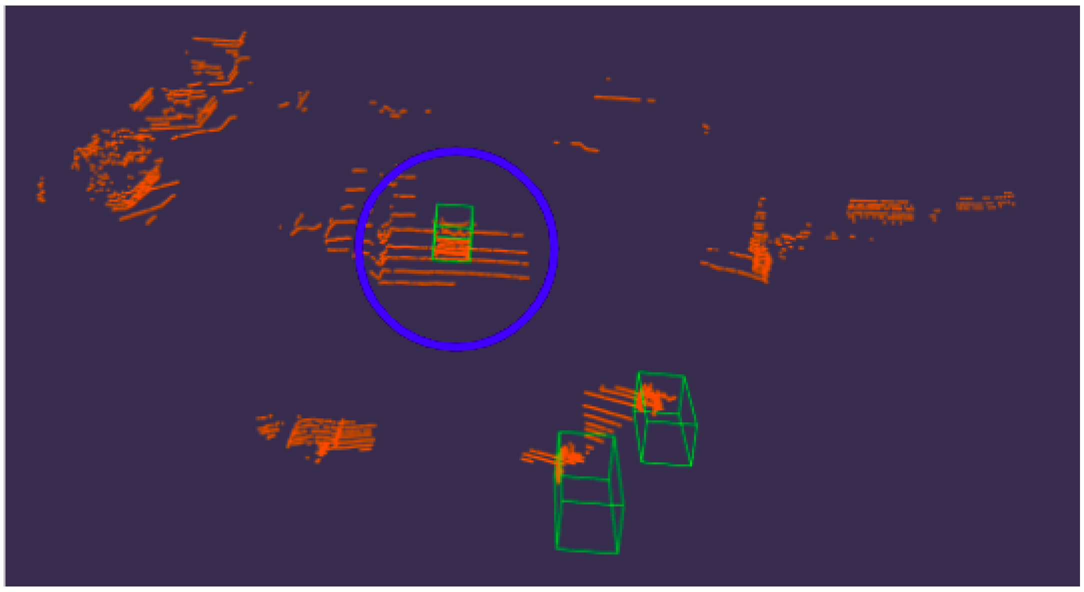
bounding box: [ 1.0739,  1.6274, 20.2643,  1.6843,  1.6877,  4.1997, -1.1749]





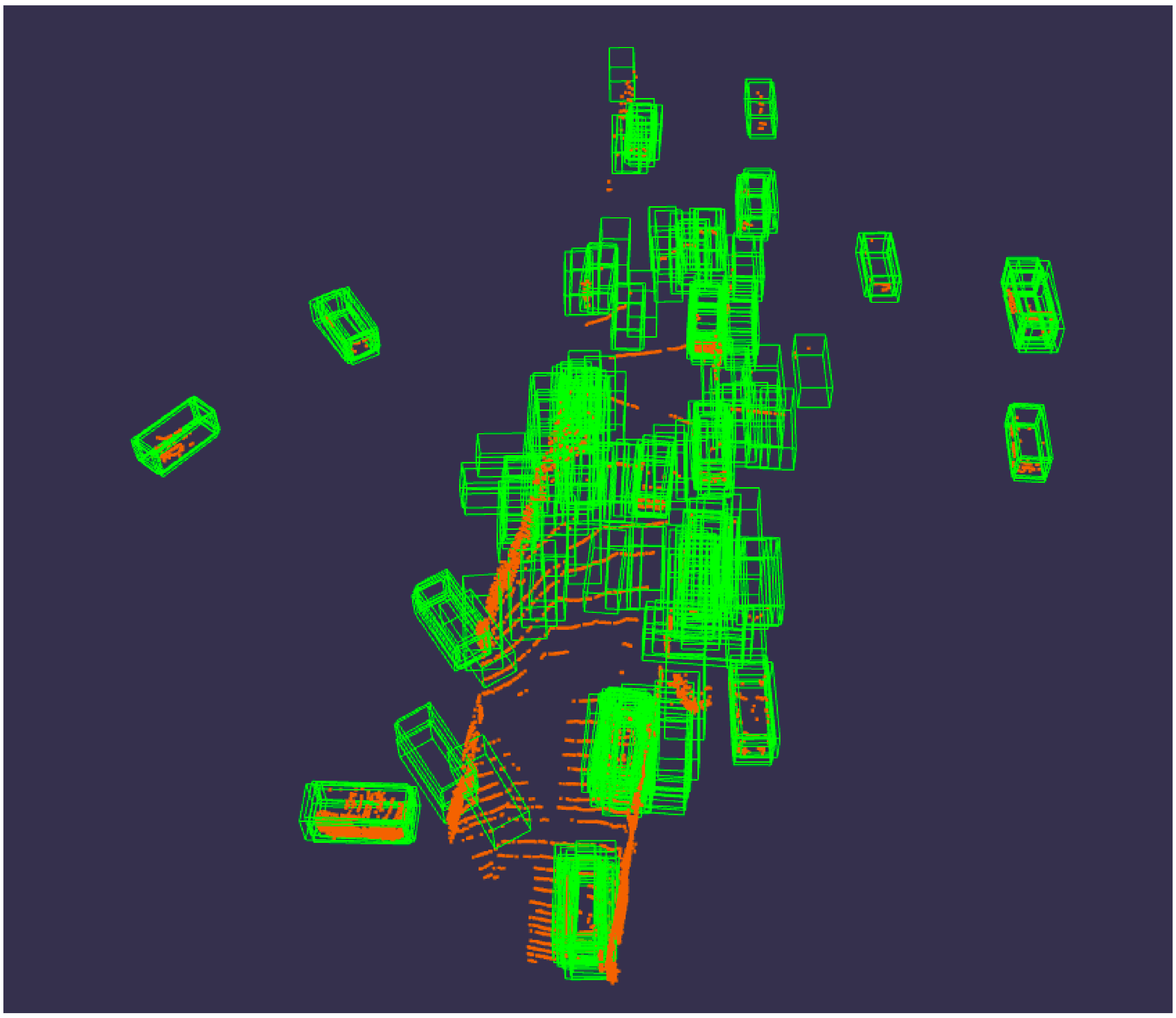
frame: DXdxO5H7RXbfVtSqBVd8

bounding box: [ 1.1498,  1.7912, 24.3467,  1.5620,  1.6330,  3.7346, -1.4909]



## 1.3 Generate the final input-output pairs per scene:

a) **Visualize a scene from the training set with all proposals. (As opposed to Fig. 1 - right your scene should only contain the pooled point. In your report, include the frame name):**



**frame: fq53kG0E7BfjATVstKvz**

b) **Why is such a sampling scheme required? What would happen if we (1) randomly sample from all proposals for each scene? (2) don't sample easy background proposal at all? (3) consider a sample to be foreground if it's above 0.5 IoU and background otherwise? Why do we need to match the ground truth with its highest IoU proposal?**

**Answer:**
We need this sampling scheme to deal with class imbalance issue.
1) If we randomly sample from all proposals for each scene, this will lead to class imbalance issue. Since  the number of objects in a scene is limited, if we randomly sample it, the number of foreground samples will be relatively small. Therefore, it is hard for the network to learn the foreground samples and reversely it will favor background samples.
2) For three categories: easy background, hard background and foreground, if we do not sample easy background at all, then the network will bias to hardground samples and its performance on predicting the easy background will be very poor.
3) The samples with IoU around 0.5 are hard to distinguish and learn. Moreover, the IOU metric only considers the intersection volume instead of considering the distinguished features; however, there might be the case that the sample with IoU equals 0.49 but it is actually fore ground by checking through their distinguished features. The same situation might also happen on background samples in another way. Therefore, introducing those samples won't really help the network to learn but deteriorate it. To avoid this, increasing and decreasing the threshold to 0.55 and 0.45 respectively for foreground and background is actually helpful.
In the last, matching the ground truth with its highest IoU proposal is based on two reasons: first, we will be more confident of using highest IoU proposal; second, since the proposal with highest IoU is more close to the ground truth, it will be easier and faster for the network to learn it and thus save some computational cost.

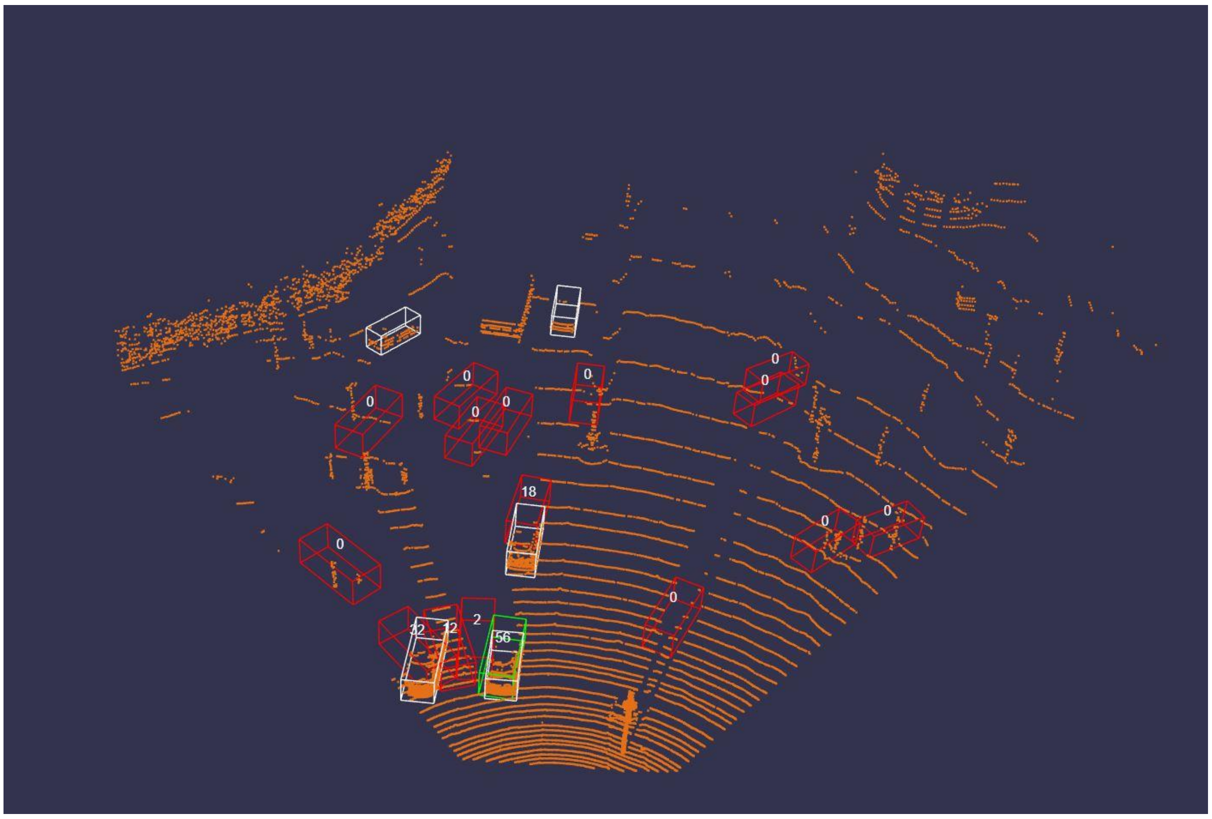
**1.5 Implement NMS with a threshold of 0.1 to reduce the number of predictions per frame. The IoU should be calculated only on the BEV.**

**Answer:**
Assuming all the cars are running on the flat ground and they are (or the bounding boxes) in the same height, then calculating on BEV or 3D does not make any changes.
However, when the bounding boxes are not in the same height, the calculation will be different: calculated on BEV will typically return a higher IOU value than calculated on 3D approach. On a 2D approach (BEV view) the intersection of a super big bounding box and a small bounding box can be the same as the intersection of two small bounding boxes; however, when we look at the 3D approach, the intersection of the former case will be smaller than the latter case.
And inside of the NMS algorithm, the IOU values we got from the 3D approach will be easily lower than the threshold t that we set. Therefore, those predictions might not be removed from the set during the looping. Finally, this will result in more final predictions than we expected. Apparently, this runs counter to the purpose of the NMS algorithm itself.

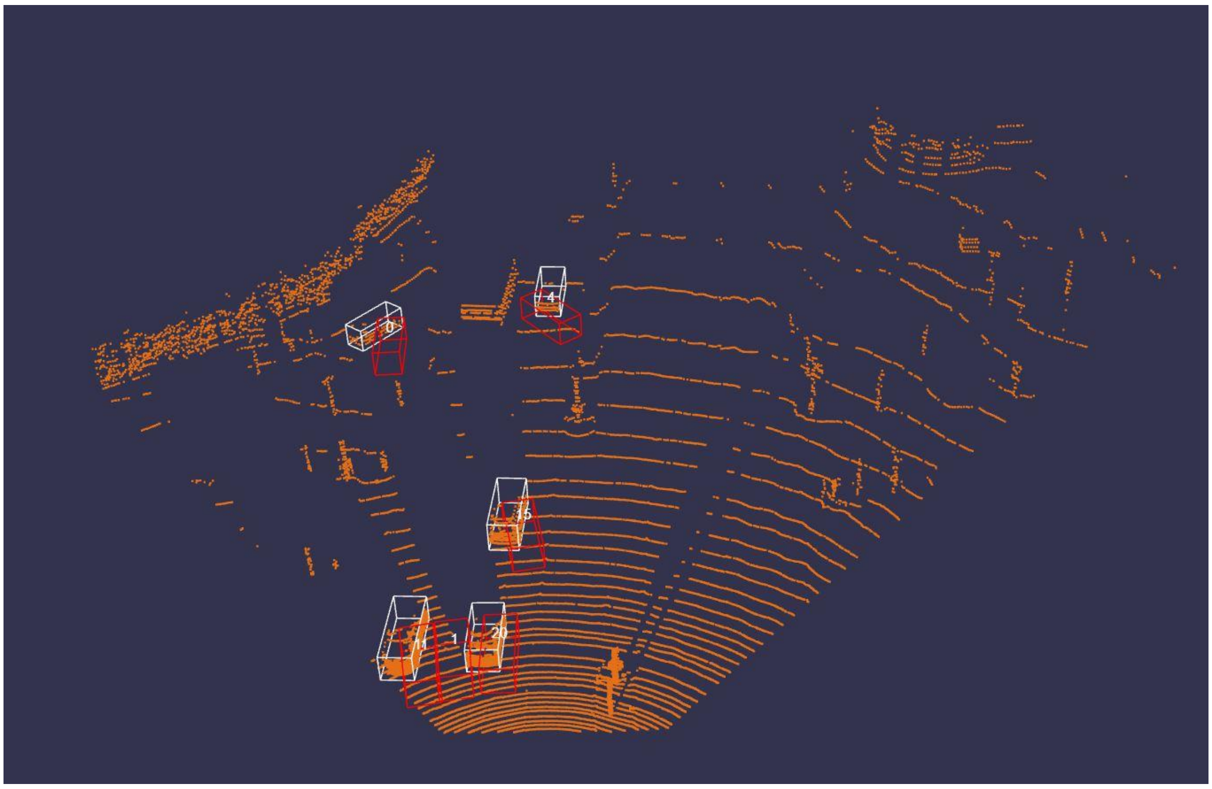
**1.6 Train the network.**

**Report: In your report, document the loss- and precision curves along with 3 example scene visualizations from the beginning, mid-way and end of the training cycle. Submit the generated submission.zip to the Codalab leaderboard to verify your implementation and report your test metrics.**

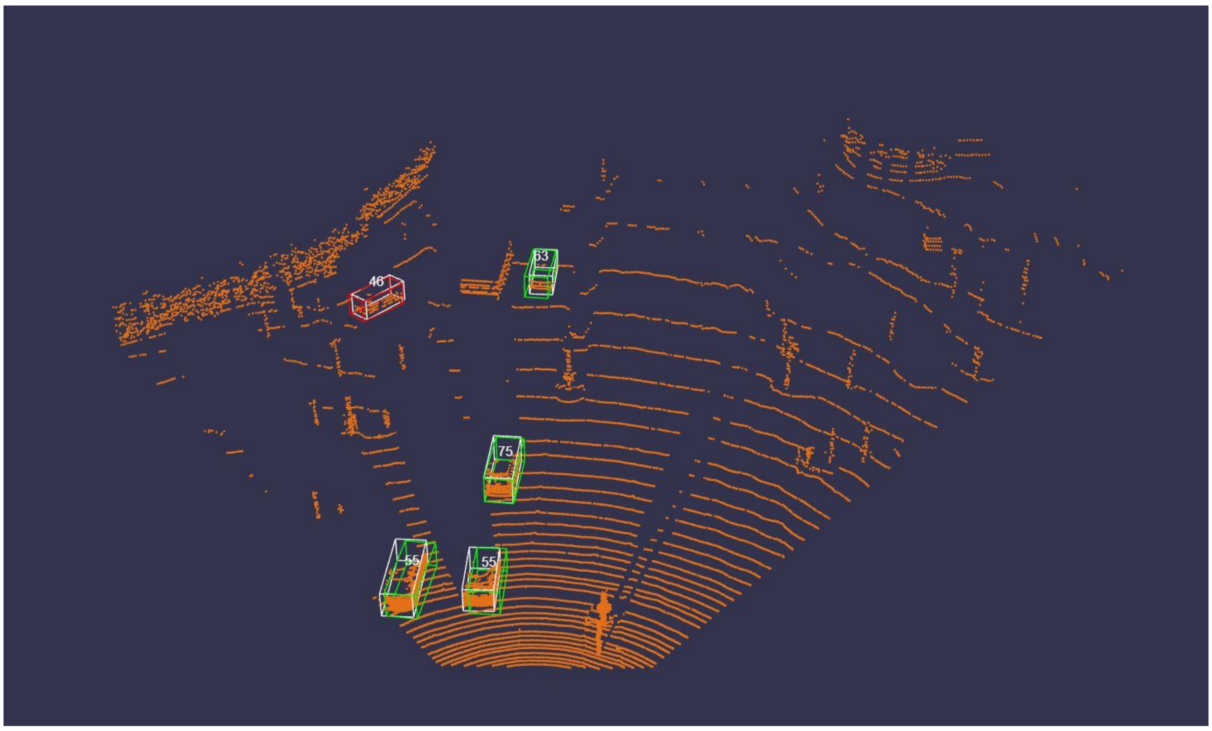
start



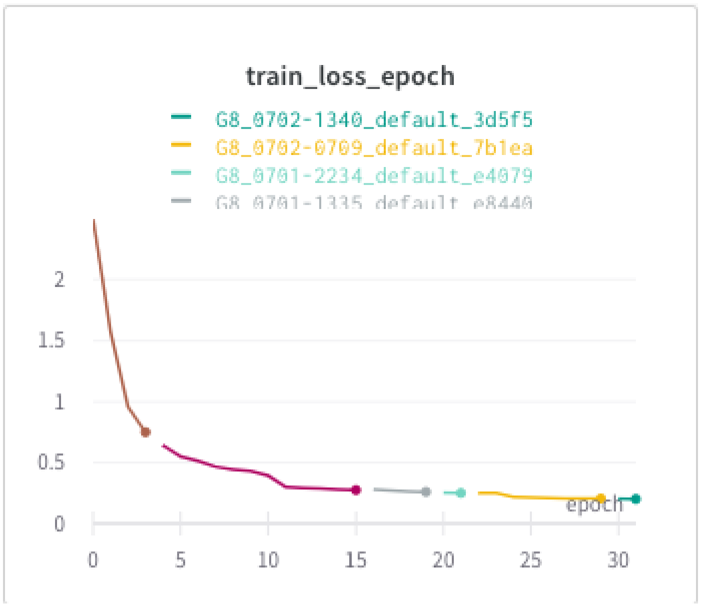
mid-way



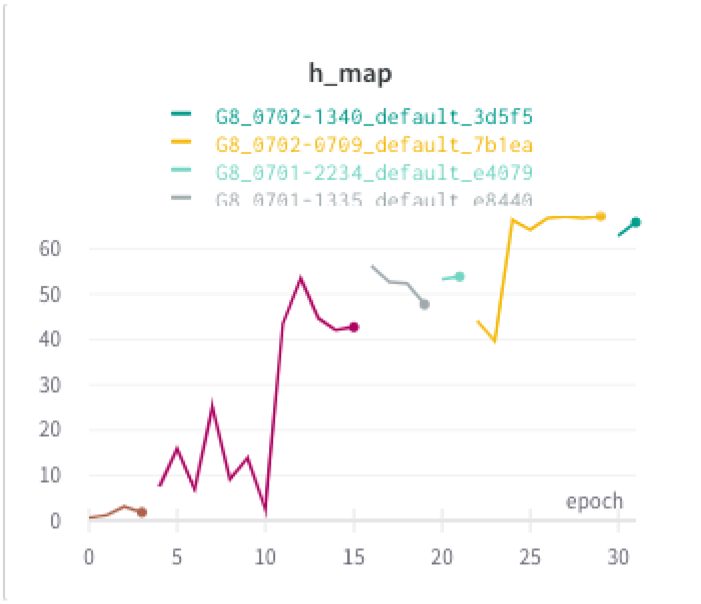
end

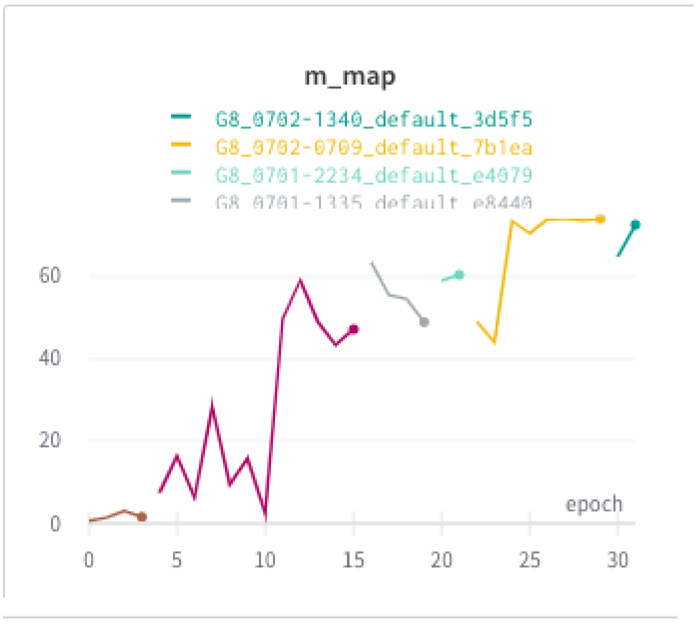


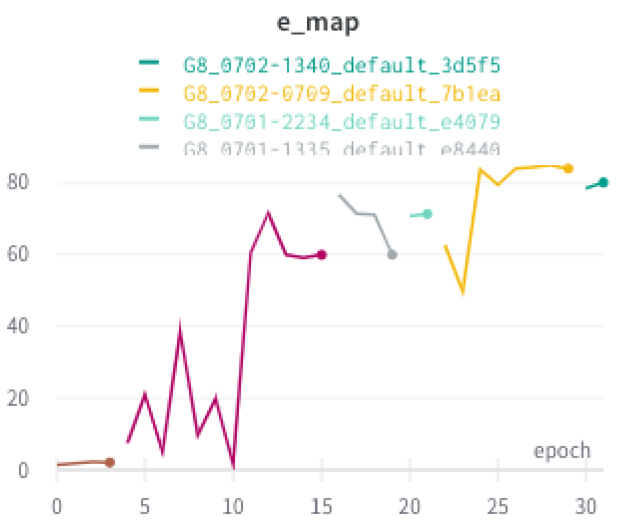
loss curve



Test metrics




m_map
G8_0702-1340_default_3d5f5
G8_0702-0709_default_7b1ea
G8_0701-2234_default_e4079
G8_0701-1335_default_e8440
60
40
20
0
epoch
0
5
10
15
20
25
30


e_map
G8_0702-1340_default_3d5f5
G8_0702-0709_default_7b1ea
G8_0701-2234_default_e4079
G8_0701-1335_default_e8440
80
60
40
20
0
epoch
0
5
10
15
20
25
30