Report 6

Force Vector Field - Design V0.1B

(MAIN = Main Pedestrian/Vehicles, OTHER = One of the other Pedestrians/Vehicles)

High Level Explanation

- Start with a vector field that follows the road and direction of MAIN (or for now maybe just the initial direction of MAIN)
- Each output = manipulation in the Vector Field by any number of OTHER
- #Outputs = "Resolution" of Vector Field
- Each Output is one point in this resolution that can be manipulated (in the future maybe one area of vectors that can be manipulated)

Implementation Explanation

- Encoder LSTM takes the same inputs (Input- and Interaction Embedding)
- At the end the hidden state is again run through a Linear Neural Network Layer. The output shape here is determined by the vector field resolution (#outputs = width * height * 2). This is implemented in modules.py (class Hidden2ForceField)
- The decoder creates a deafult force field for each pedestrian where each vector points in the direction of the past movement of MAIN. (In the future, this could instead encode the flow of the road). Then each output of the encoder has the ability to change one vector in the vector field. Then the current position of MAIN is taken and, with the help of interpolation, the current force vector is calculated. This vector should then be integrated over time, thus being velocity that can be added to the last predicted position.

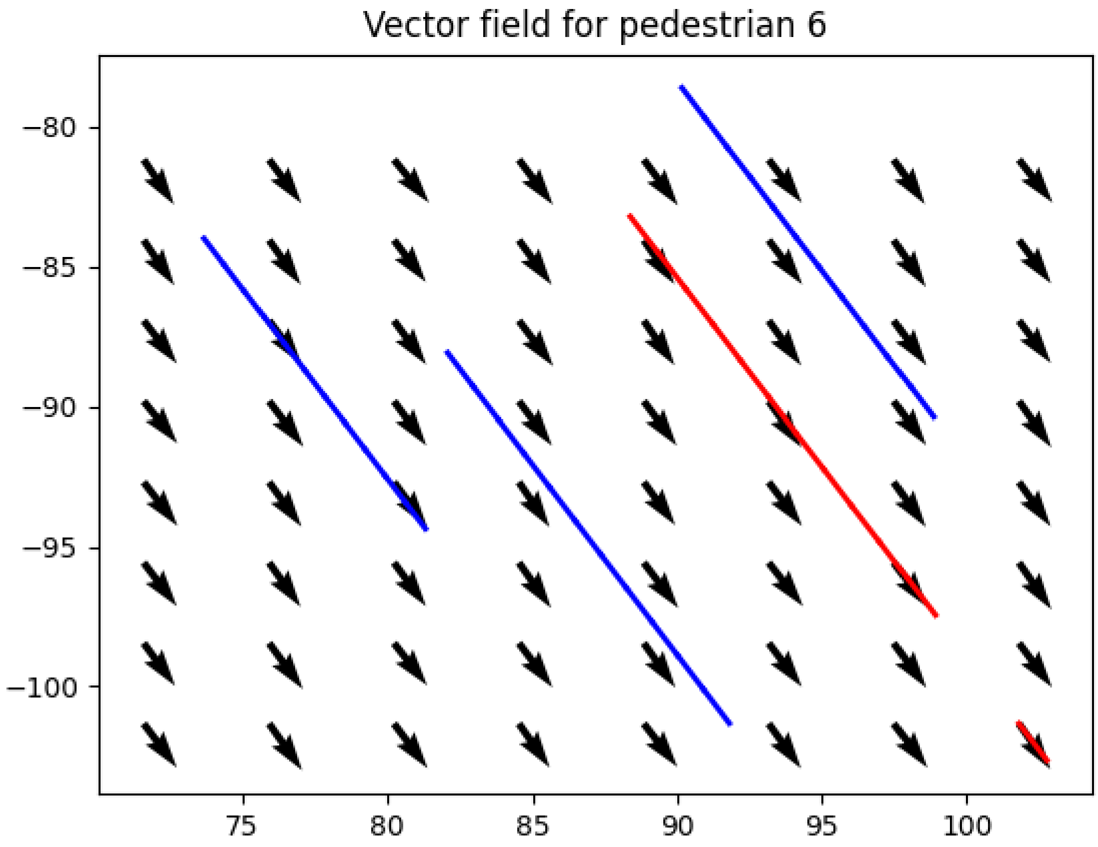
Notable Implementation Details of the original LSTM:

- The encoder predicts the first step of the prediction (?)
- The decoder (and encoder on the first step) calculates predicted velocities of pedestrians as a multivariate normal and then uses only the mean to update the position.
- Following Image (and the actual scenes) are rotated due to augmentation of the scenes for improved learning

Vector Field during first epoch of training without manipulated vectors

- MAIN is red

- MAIN line is observed, MAIN dot is predicted position



Currently working on:

- Fixing Loss function (result of Loss is NAN. Their loss was based on the multivariate normal, thus i have to make my own)

Next Steps:

- Properly integrating force
- Manipulating areas of vectors instead exact vectors in vector field
- Training and evaluating force vector field encoder