Motion Prediction Challenge on Waymo Open Motion Dataset

by

Gudiwada Raghava

Regd. No: 212IS009

under the guidance of

Dr. Basavaraj Talawar

Department of Computer Science & Engineering,
National Institute of Technology Karnataka, Surathkal,
Mangalore, India - 575025.

May 26, 2023

Table of Contents

Table of Contents

Introduction

- The World Health Organization estimates that 1.3 million people die from traffic accidents every year.
- A majority of these accidents are avoidable, a product of human error.
- With the recent development of autonomous vehicles (AVs), it seems possible to drastically reduce the number of accidents and save lives. In order to do so, and to be able to improve the safety of AVs, one important task that has attracted a lot of attention is to be able to accurately predict the motion of nearby objects.

Table of Contents

Motivation

- Potential to enhance safety
- Optimize planning
- Decision-making processes
- Improve traffic flow
- Facilitate data analysis and visualization
- By accurately predicting the positions of multiple agents in the future, various industries and research fields can benefit from improved efficiency, safety, and user experiences.

Table of Contents

Literature Survey

AUTHOR	YEAR	ARCHITECTURE	Proposed Methodology
Kirill Brodt, Artsiom Sanakoyeu	2021	MotionCNN, Xception71	The model consists of CNN backbone pretrained on ImageNet with one fully-connected layer attached on top. The model takes a multi-channel raster image as input and predicts K trajectories along with the corresponding confidence values c1, . . . , cK.
Stepan Konev	2022	Multipath++	Experimented with a single decoder with 6 modes and 5 decoders each with 6 modes followed by attention mechanism and multi-context gating block (MCG) from MultiPath++ that mapped 30 modes into required 6. With multiple decoders we used a proposed strategy of blocking weighs update for randomly selected decoders.

Literature Survey (Contd..)

AUTHOR	YEAR	ARCHITECTURE	Proposed Methodology
Elmira Amirloo, Amir Rasouli, Peter Lakner	2018	Generative Adversarial Network (GAN), Multi-Agent Transformer, Graph-based model	The proposed model consists of three main modules: trajectory encoder, multi-resolution map encoder and multi-agent decoder.
Yicheng Liu, Jinghuai Zhang, Qinhong Jiang	2019	Long Short-Term Memory(LSTM), Graphical Neural Network(GNN)	Proposed a novel end-to-end multimodal motion prediction framework called MultiModal Transformer (mmTransformer).
Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun	2020	Target-driveN Trajectory (TNT) and the DenseTNT	The modular approach described in the TNT and DenseTNT works with a simpler convolutional approach that was able to perform on par with more complex, state-of-the-art models to predict future object in a path.

Table of Contents

Problem Statement

Problem Statement

Given agents' tracks for the past 1 second on a corresponding map, predict the positions of up to 8 agents for 8 seconds into the future.

Table of Contents

Objectives

Objectives 1

Building the model for the motion prediction tasks, LSTM Goal prediction model is to estimate the endpoint of objects 8 seconds in the future provided with 1 second of past context. This model is used to find the FDE metric.

Objectives 2

The Trajectory completion model receives as input a preprocessed scenario and couples it with the estimated endpoint given from the goal prediction model. The goal of this model is to predict 80 coordinates, ten for each second in the future, for each object conditioned on their goal. This model is used to find the ADE metric.

Objectives (Contd..)

Objectives 3

The combination of two models is used to evaluate performance by requiring models to predict trajectories for up to eight objects in a given scenario and also finding the ADE & FDE metric for motion prediction challenge.

Table of Contents

Preprocessing Data

- The pre-processing step includes a normalization step where features are centered about the origin so as to stabilize training.
- Feature extraction is one of the main routes of experimentation in our project design, our pre-processing step rasterizes as seen in Figure 1 each segment's context and determines what features are most relevant for each model.
- Specifically in our work, we included most object and roadgraph features, choosing to omit less pertinent features such as object size and sparse roadsign features such as traffic lights and their states.

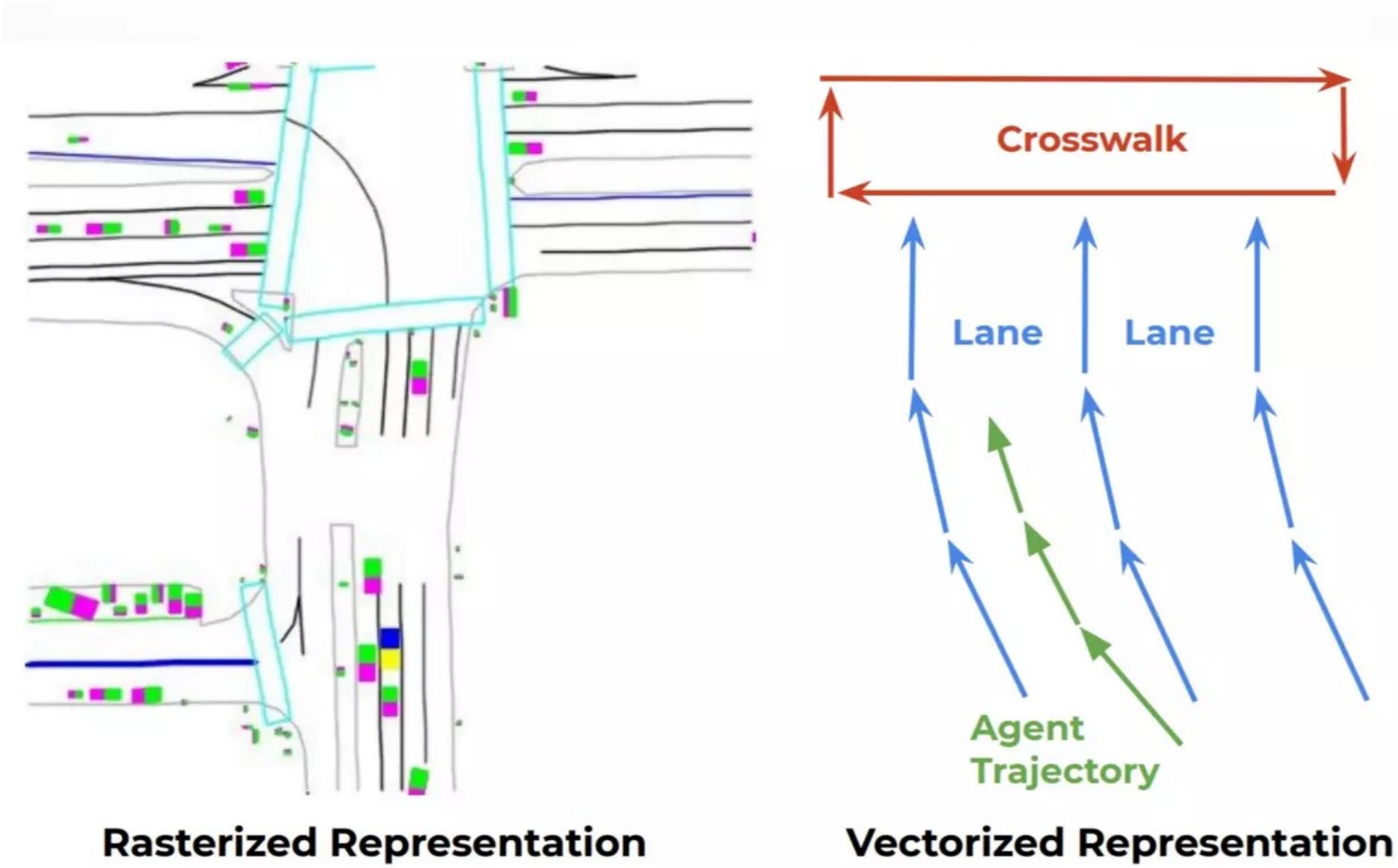


Figure 1: Vectorized data representation

Model overview

- We chose to only output a single goal and trajectory for each object, omitting an additional trajectory ranking module.
- The final model consists of two modules:
  1. Goal prediction module
  2. Trajectory prediction module.
- We extract relevant features as described in pre-processing step and vectorize the provided information for use as the input to each model in our pipeline.
- This approach allows us to identify and select features that are more relevant to a given module, reducing irrelevant information and the complexity of our overall model.
- The motion prediction task, as specified by the Waymo challenge, evaluates the performance by requiring models to predict trajectories for up to eight objects in a given scenario.
- Therefore, both goal and trajectory modules generate outputs for eight different objects at a time in a given scenario.

Long Short-Term Memory

- Long Short-Term Memory (LSTM) blocks are usually used in sequential/temporal applications and their structure can be seen in Figure 2.
- Their recurrent structure makes it possible to better learn sequences of data, with the memory component storing pertinent information that can be referenced later in a sequence.
- Since our model contains 10 discrete states containing past context and 1 discrete state containing current context, which are temporally related, LSTM blocks seem to be a strong approach that can capture these dependencies.
- As seen in Figure 2, the block in the middle does not only take information about the current time step (X_t) but it also retrieves information from all previous steps ($X_1, X_2, ..., X_{t-1}$).
- Hence, the outputs at time t (Y_t) depend on all the previous inputs in addition to the current input.

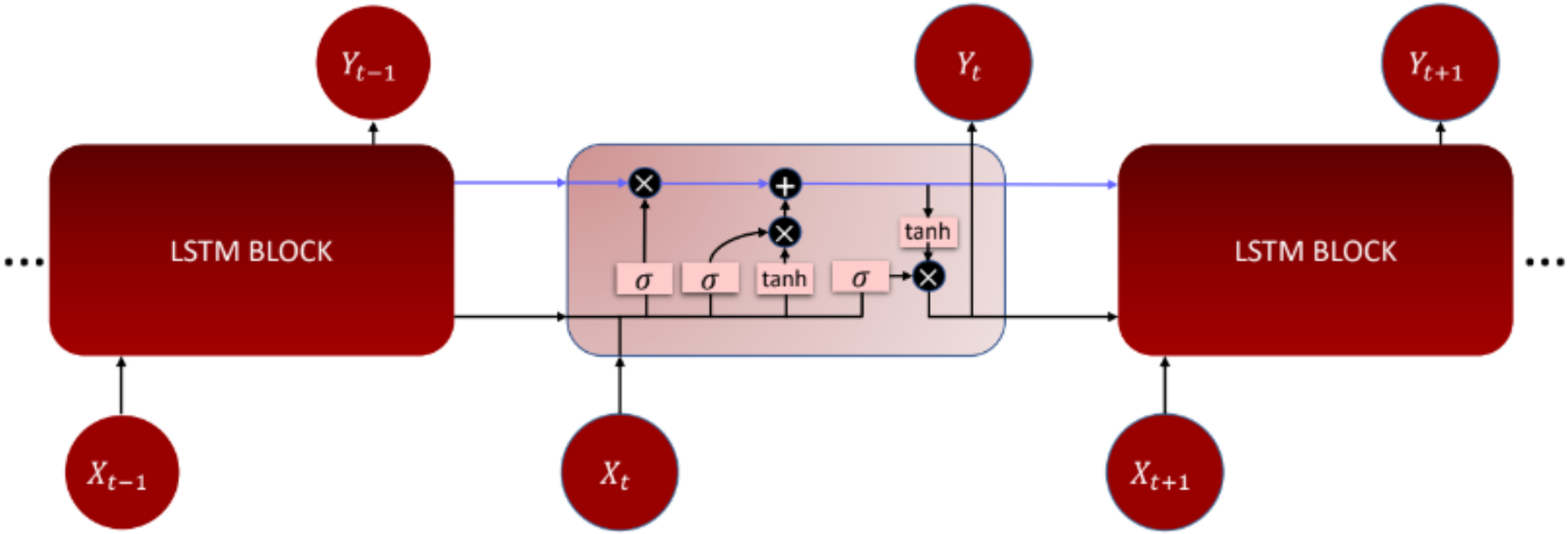


Figure 2: Internal LSTM cell structure.

Goal Prediction

- The principle behind our goal prediction model is to estimate the endpoint of objects 8 seconds in the future provided with 1 second of past context.
- These estimates will then be used as anchors to predict object trajectories in the subsequent model.
- Though we introduced the LSTM chain above as having inputs and outputs in every block, we used an alternate structure in our goal prediction and trajectory completion modules.
- To estimate target positions 8 seconds into the future, we used the model provided in Figure 3.
- The model can be deconstructed into a time-based component and a static component. The time-based LSTM chain sequentially processes inputs containing past states (including object velocity, yaw, and position).
- The chain contains eleven blocks (ten for the 1 second of past context, one for the current state), and only the output from the last block is used in the final dense layer.

Goal Prediction (contd..)

- In addition to the time-dependent object information, each scene also contains static information such as the underlying roadgraph and other time-independent features.
- These features do not change over each timestamp and are instead processed independently.
- Static features are fed through four convolutional layers, each followed by a max pooling layer.
- To decrease the overfitting, we also add a dropout layer here. The output of the dropout layer is concatenated with the time-dependent outputs from the LSTM block.
- The resulting flattened vector is then fed into a dense layer and the output of the last dense layer is used as the endpoint predictions for all the vehicles we need to predict.

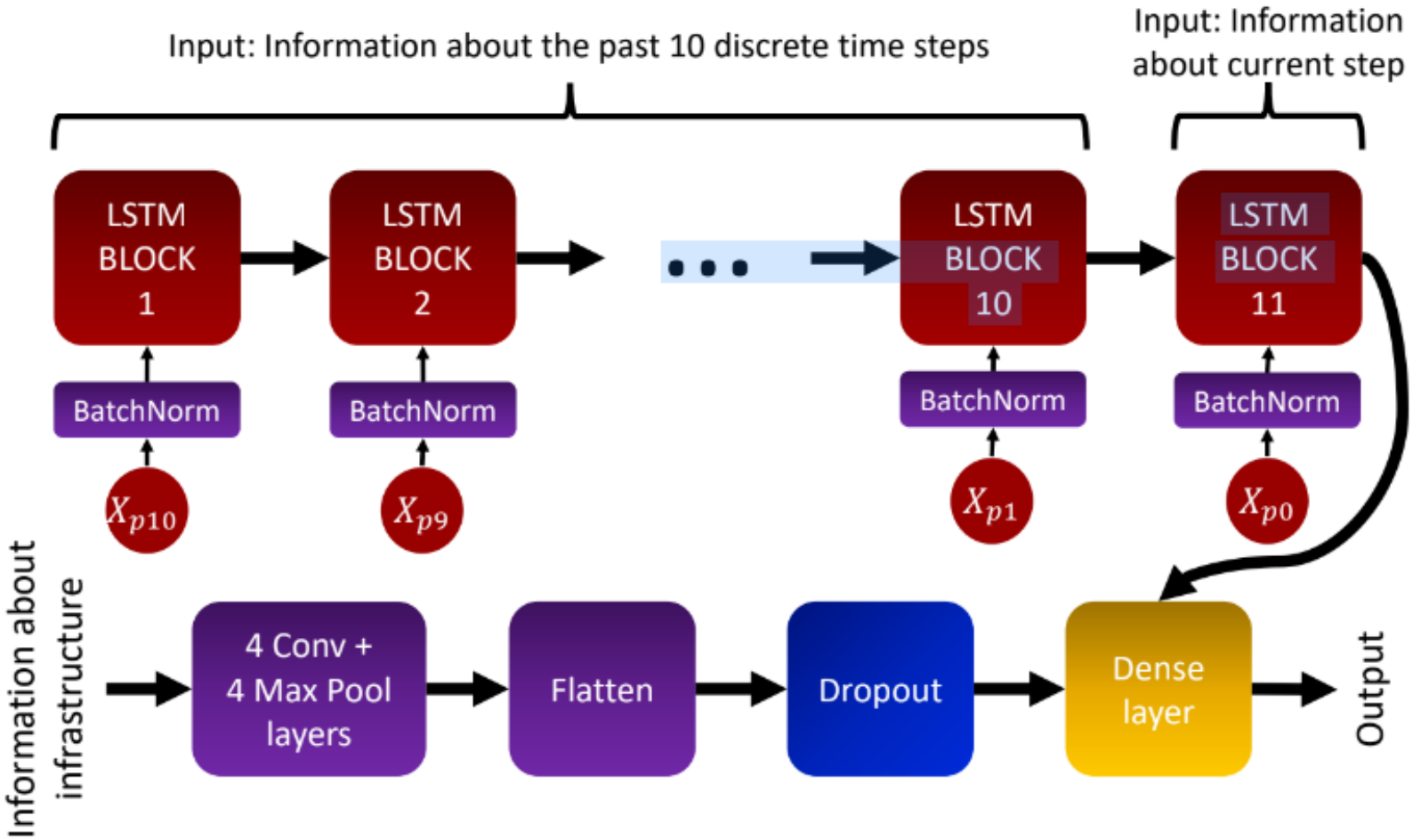


Figure 3: End point prediction model.

Trajectory Completion

- The trajectory completion model receives as input a preprocessed scenario and couples it with the estimated endpoint given from the goal prediction model.
- The goal of this model is to predict 80 coordinates, ten for each second in the future, for each object conditioned on their goal.
- The first half of this model is similar to the goal prediction model given in Figure 3 (the 'encoder'). Similar to the goal prediction model, time-dependent features are processed independently of static features prior to use in the decoder.
- The only difference between our previous model and the trajectory encoder is that encoder inputs now also contain the broadcasted predicted end points.

Trajectory Completion (Contd..)

- To predict the future points, we also define a 'decoder', given in Figure 4.
- The decoder contains 8 LSTM cells, with each cell responsible for predicting one second of future motion.
- This architecture is similar to the cell structure described in the Long Short-Term Memory in Figure 2 with inputs being 0.
- The outputs of each LSTM should, when passed through a dense layer, output 10 points for each object, corresponding to one second of the eight-second prediction window. With all eight cells, we obtain the full 8-second predicted trajectories for all target objects.

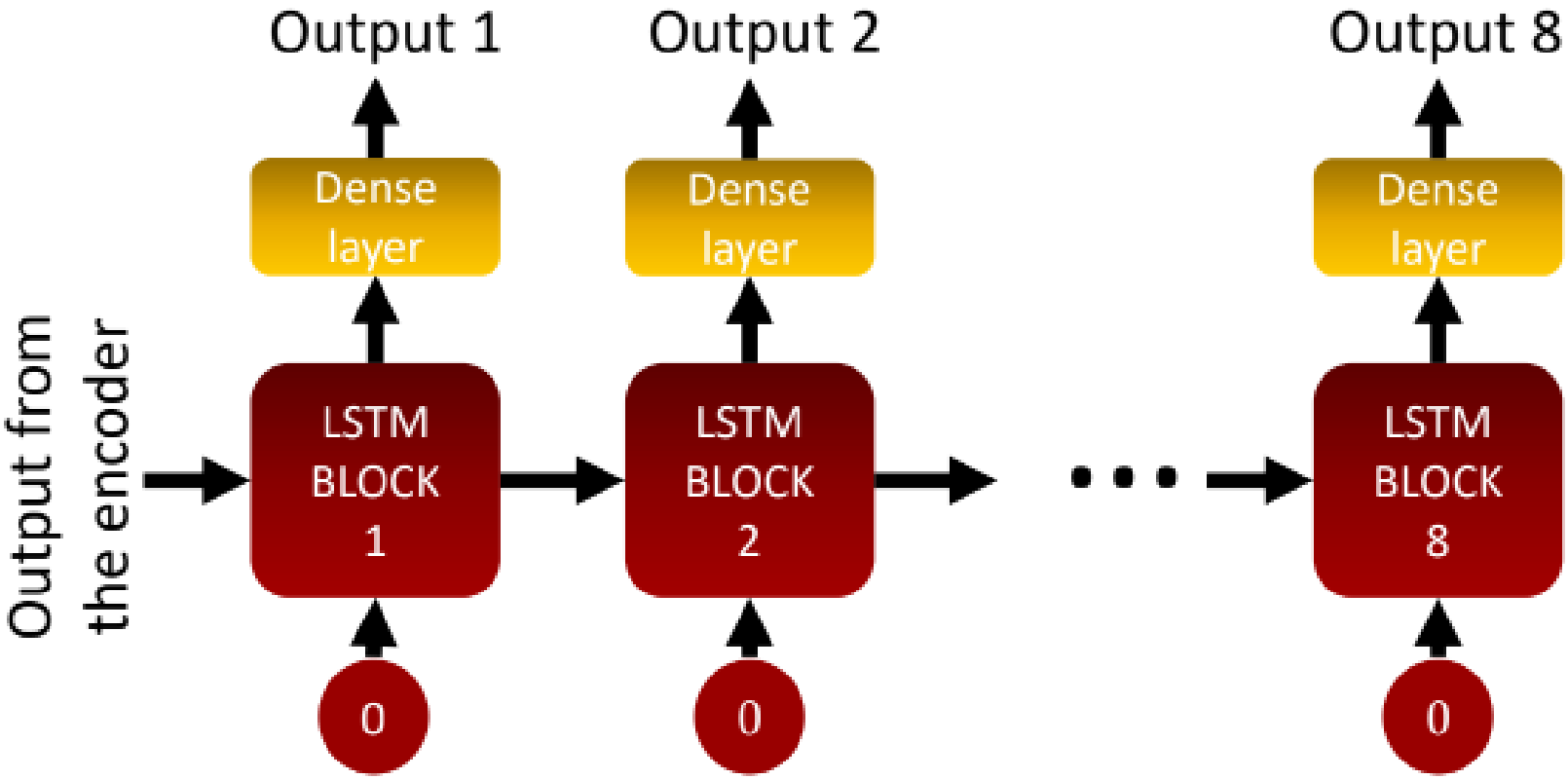


Figure 4: Trajectory prediction model.

Table of Contents

Dataset

- The dataset consists of 103,354 segments, each containing 20 seconds of object detection sampled at 10 Hz.
- The segments are divided into 9 second long segments with a 5 second overlap, which creates a total of 310,062 segments.
- Each segment has a corresponding scene containing information about the surroundings such as the roads, lanes, and sidewalks, object states and traffic light states.
- Furthermore, each segment contains pertinent information on up to 128 different objects in the scenario, including their type, size, position, velocity, yaw, among other features.
- Partially observed objects are marked as such in the scenario data. One example of one scene at two different time points can be seen in Figure 5 & 6.

Visualization

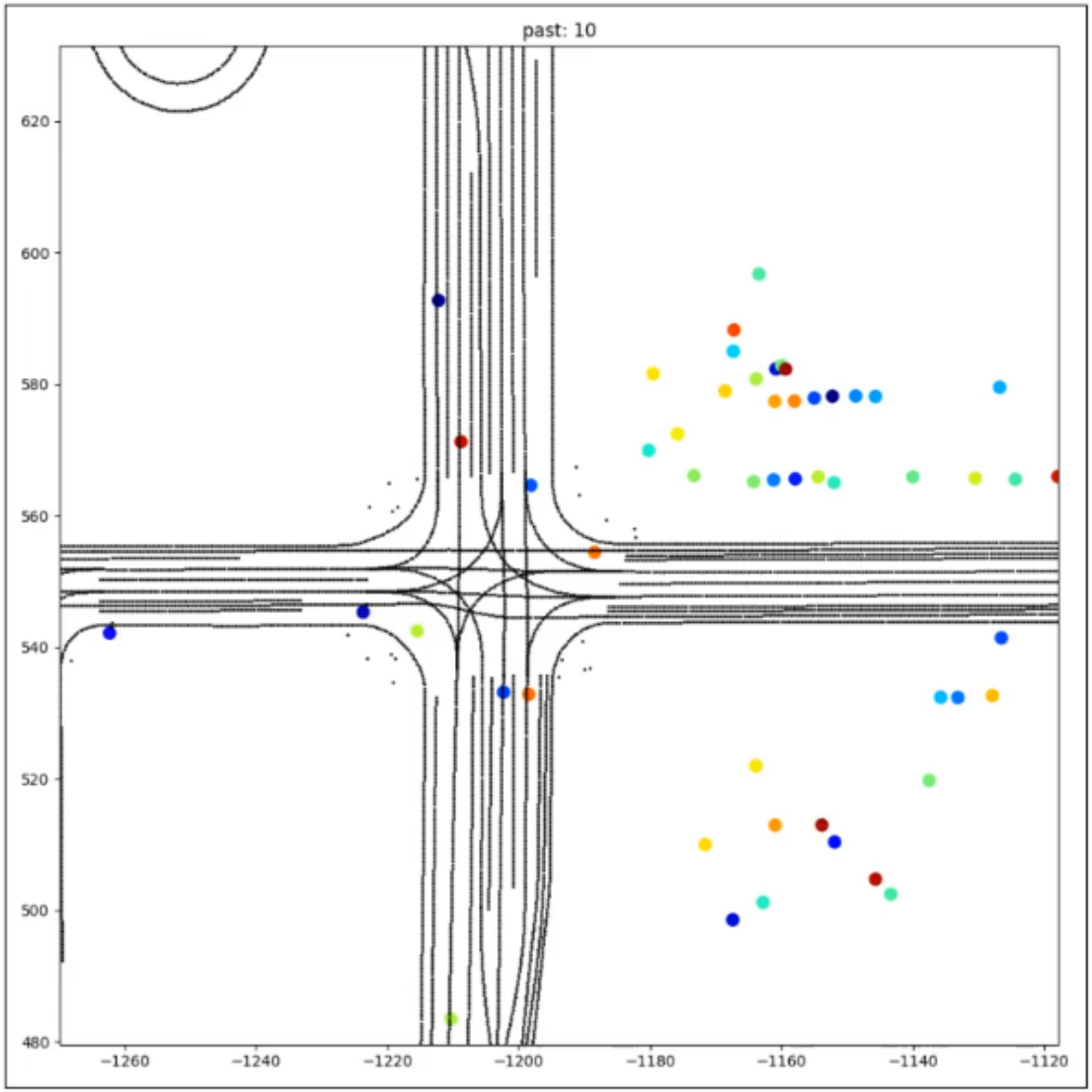


Figure 5: Scene at time step 0.

Visualization (Contd..)

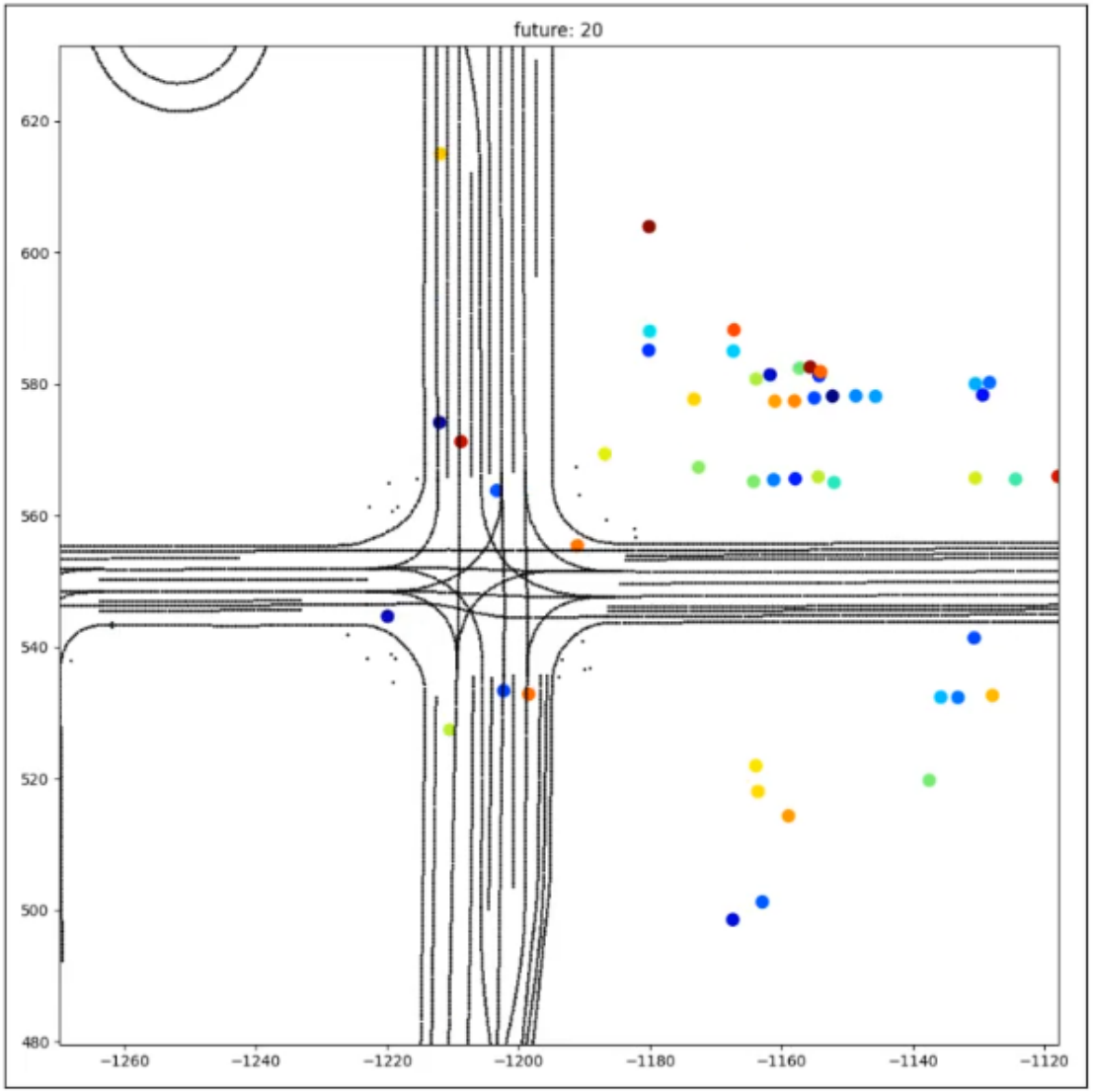


Figure 6: Scene at time step 20.

Specifications

Hardware Specifications

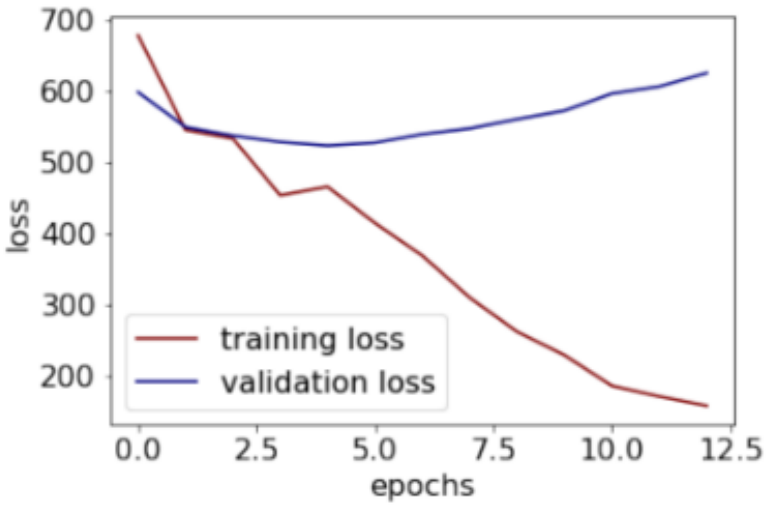
- Jupyter Notebook Environment or Google Colab
- Atleast 16GB RAM is required
- 250GiB System Memory
- A high-end GPU with CUDA support is crucial for accelerating the training and inference processes.
- NVIDIA Tesla V100 32GB GPU

Software Specifications

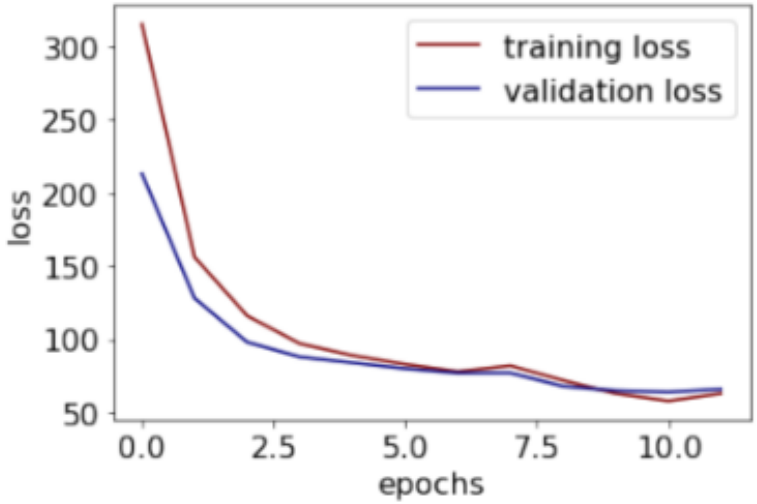
- **Language** : Python
- Tensorflow 2.11, numpy==1.21.5, matplotlib==3.6.1, pandas==1.5.3, pillow==9.2.0, openexr==1.3.9, torch, opencv-python, protobuf $<$ 3.20, tqdm $\geq$ 4.45.0, and Keras 2.6 these all libraries are required.

Table of Contents

Model Validation



(a) Loss for the goal prediction model.



(b) Loss for the trajectory completion model.

Figure 7: The loss functions during training of the two models.

Table of Contents

Results

Table 1: Hyperparameters for the two modules.

Goal Prediction Module		
Learning Rate	**Batch Size**	**Dropout**
0.001	128	0.4

Trajectory Completion		
Learning Rate	**Batch Size**	**Dropout**
0.001	128	0.4

Metrics for Model Evaluation

Average Displacement Error (ADE), Final Displacement Error (FDE) are the commonly used metrics for evaluation:

$$ADE = \frac{1}{T}\|X^{\text{gt}} - X\|_2 \tag{1}$$

$$FDE = \|X_T^{\text{gt}} - X_T\|_2 \tag{2}$$

- "X^{gt}" represents the ground truth trajectory.
- "X" represents a predicted one.
- "X_T^{gt}" represents the ground truth trajectory at time T, and
- "X_T" represents the predicted trajectory at time T.

Metrics for Model Evaluation (Contd..)

To evaluate multiple hypotheses we use Minimum Average Displacement Error (minADE) and Minimum Final Displacement Error (minFDE):

$$minADE = \min_k \frac{1}{T} \|X^{\text{gt}} - X_k\|_2 \tag{3}$$

$$minFDE = \min_k \|X_T^{\text{gt}} - X_{k,T}\|_2 \tag{4}$$

- "k" represents the index of the trajectory.
- "X_k" represents the predicted trajectory for the corresponding index k, and
- "$X_{k,T}$" represents the predicted position of the trajectory at time T for the hypothesis with index k.

Results

Table 2: Evaluation on train and validation sets of Waymo Open Motion Dataset.

Goal Prediction Model			
Dataset	**loss**	**Min ADE**	**Mean FDE**
Train	157.04	12.5	14.73
Valid	523.76	22.8	24.57

Trajectory Completion Model			
Dataset	**loss**	**Min FDE**	**Mean FDE**
Train	58.29	7.6	8.5
Valid	63.90	8.2	10.15

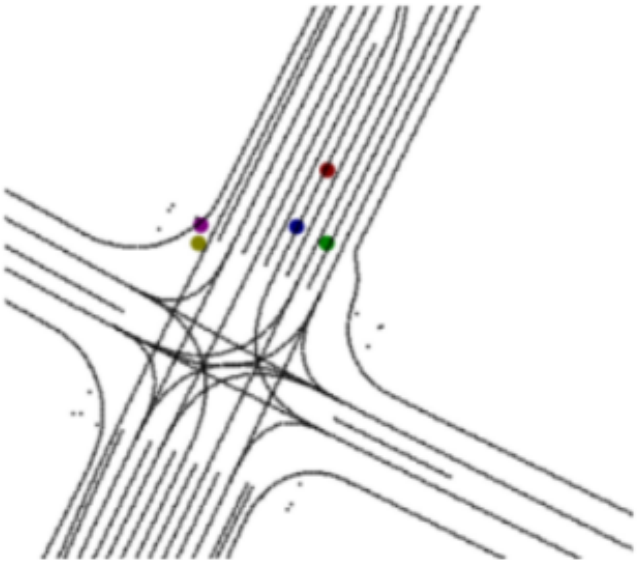
Results

Table 3: Compararive analysis on test dataset of Waymo Open Motion Dataset.

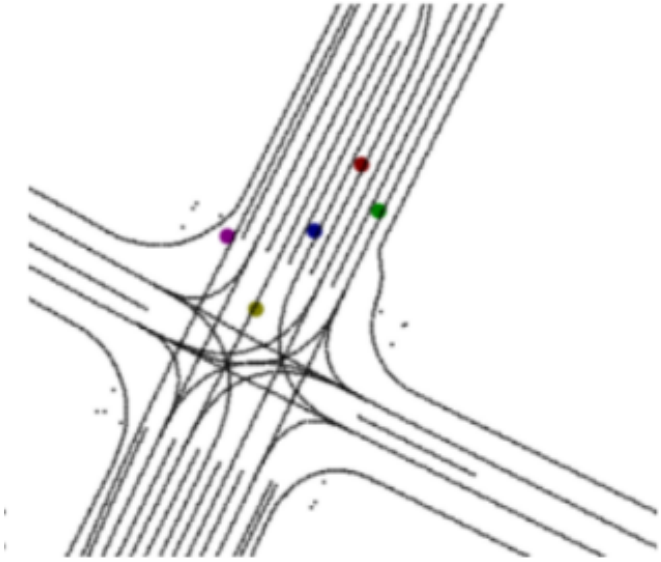
Combined Model		
	Mean ADE	**Mean FDE**
Test	13.15	23.95

Research paper CNN Model		
	Mean ADE	**Mean FDE**
Test	22.32	29.58

Results



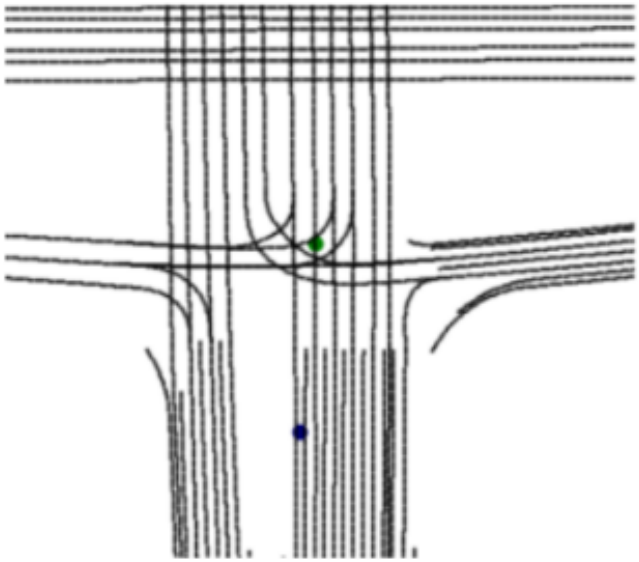
(a) GT for training example.



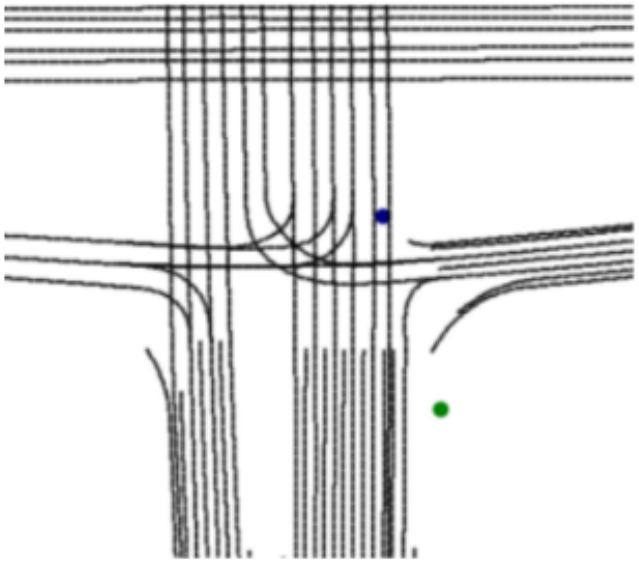
(b) Prediction for training example.

Figure 8: The ground truth (GT) positions and the predicted positions for a training example.

Results



(a) (c) GT for validation example.



(b) (d) Prediction for validation example.

Figure 9: The ground truth (GT) positions and the predicted positions for a validation example.

Table of Contents

Conclusion

- We partitioned the task of motion prediction into two subtasks: goal prediction and trajectory completion.
- Implementation of the goal prediction model had a very high variance and overfitted to the training data, which decreased the overall performance of the system.
- One way to improve upon on these results is to increase the regularization or the training data to reduce the overfitting of the goal prediction module.
- Implementation of trajectory completion model had a far better performance and seems promising as a trajectory completion model.
- However, due to the low performance of the goal prediction model, our overall model's ADE and the FDE is 13.15 and 23.95 are respectively.

Table of Contents

Future Work

- The future work include evaluating other model architectures using 80 LSTM cells, one for each timestamp.
- Further tuning the hyperparameters of the models with a greater focus on the goal prediction model, and evaluating the incorporation of other static and time-dependent context features.

References I

3. World Health Organization. Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, (accessed Nov 26, 2022)
4. Waymo. Waymo Open Dataset. URL:https://waymo.com/open/download/, 2022.
5. F.A. Gers and E. Schmidhuber. LSTM recurrent networks learn simple context-free and context-sensitive languages. IEEE Transactions on Neural Networks, 12(6):1333–1340, 2001.
6. Gabor Melis, Tomas Kocisky, and Phil Blunsom. Mogrifier LSTM. arXiv preprint arXiv:1909.01792, 2019.
7. Kirill Brodt, and Artsiom Sanakoyeu. Motioncnn: A strong baseline for motion prediction in autonomous driving. In Workshop on Autonomous Driving, CVPR, 2021.

References II

8. Junru Gu, Qiao Sun, and Hang Zhao. Densetnt: Waymo open dataset motion prediction challenge 1st place solution. arXiv preprint arXiv:2106.14160, 2021.
9. Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. arXiv preprint arXiv:2008.08294, 2020.
10. Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11621–11631, 2020.

References III

11. Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8748–8757, 2019.
12. Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9710–9719, 2021.

References IV

13. Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. arXiv preprint arXiv:1910.03088, 2019.

Thank You !!