

Trajectory Prediction Based on Planning Method Considering Collision Risk

Ya Wu¹, Jing Hou¹, Guang Chen^{1,2} and Alois Knoll²

Abstract—Anticipating the trajectory of Autonomous Vehicles (AV) plays an important role in improving its driving safety. With the rapid development of learning-based method in recent years, the long short-term memory (LSTM) network for sequential data has achieved great success in trajectory forecasting. However, the previous LSTM only considered forward time cues and did not reason on motion intent of rational agents. In this paper, we use planning-based methods follow a sense-reason-predict scheme in which agents reason about intentions and possible ways to the goal. In addition, the collision risk is considered, and the most appropriate future trajectory will be selected with the current state of the agent. We have compared our method against two baselines in highD dataset. Our experimental results show that the planning-based method improves prediction accuracy compared with the baselines.

I. INTRODUCTION

Autonomous vehicles in the near future is expected to reduce the number of road accidents and improve driving safety [1]. However, in order to reach the level of human drivers, more and more researchers begin to focus on anticipating their future behaviour (a.k.a motion or trajectory).

For trajectory prediction methods, we can divide them into three categories: traditional methods (constant velocity, Kalman Filter [2]), learning-based methods (Mixture Density Networks [3], Convolutional Neural Networks [4], Long Short Term Memory [5]) and the planning-based methods [6].

Due to the traditional method is simple and easy to implement, many early researchers used this method to predict the trajectory of autonomous vehicles, and these methods are only effective in simple driving scenarios. Christoph Scholler [7] believed that the simpler the model, the better performance of this method. They proposed a simple Constant Velocity Model which can outperform even state-of-the-art neural models. They thought the interactions are too complex to predict, but interactions among vehicles, or pedestrians and vehicles, may be much more predictable as vehicles move in highly structured environments.

Most recently, learning-based approaches have become more and more popular due to their superior performance in complex environments compared to the conventional approaches. Yeping Hu [3] proposed a semantic based intention and motion prediction (SIMP) method, which can generate

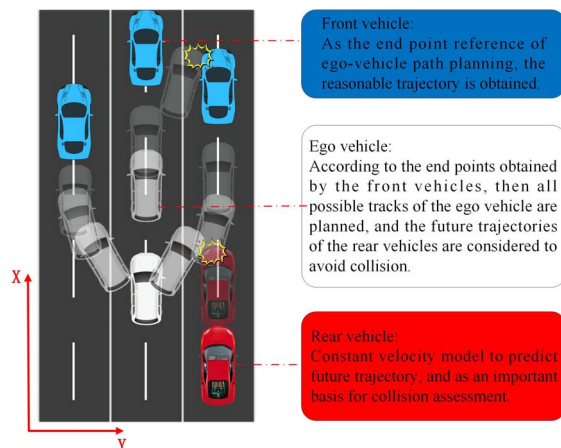


Fig. 1. System diagram, white vehicle represents ego vehicle, blue vehicle represents front vehicle, red vehicle represents rear vehicle.

the probability distribution of the designed semantic description given a certain representation of the current state, and then assign a mixture density network (MDN) based on Gaussian mixture model (GMM) to each insertion region. It can be adapted to any driving scenarios, but this method does not consider the rationality of trajectory and collision risk.

Planning-based methods explicitly reason about the motion intention of agents, according to the summary of Rudenko et al. [8], the planning-based approaches can be classified into two categories. On the one hand, Vasquez et al. [9] used a pre-defined reward function, with the planning technique (Fast Marching Method), then an efficient goal estimation and full spatiotemporal prediction algorithm was developed with lower complexity than comparable approaches. And the results are more accurate than the current state-of-the-art approach. On the other hand, they estimate the reward function or action model with the inverse planning methods. Previtali et al. [10] proposed a standard Inverse Reinforcement Learning (IRL) technique and estimated a reward function within a Markov Decision Process (MDP) model, and used this reward function to estimate the agent's motion in future.

In this paper, we propose a planning-based methods to anticipate the trajectory of autonomous vehicles. First, we select ego-vehicle as the target vehicle to predict its trajectory. Similar to [3], we obtain all drivable areas as target points in real time at each instant, and then plan the trajectory set to all target points. Then, we consider the collision risk

*Guang Chen is the corresponding author.

*This work was supported by the Youth Program of National Natural Science Foundation of China (Grant No. 61906138) and Shanghai AI Innovation Development Program 2018.

¹Tongji University, Shanghai, China;

²Technical University of Munich, Munich, Germany.

between the planning trajectory and the surrounding vehicle, which is inspired by [11], For the surrounding vehicle, we use the constant velocity model to predict its trajectory and get the collision risk with the planning trajectory of the ego vehicle. In addition, according to the description in [7], too many historical information have a negative impact on the generalization ability of the neural network. Referring to the method, we consider using the state of the current frame and the previous frame of the ego vehicle to choose the best planned trajectory.

The key contributions of our work are as follows:

- Trajectory forecasting method based on planning is proposed, compared to [3], the vehicle dynamics and road geometry are considered to obtain the drivable area.
- The ego vehicle planning trajectory considers the risk of collision with the surrounding vehicle.
- Reducing the algorithm complexity and improving the calculation speed.

II. PROBLEM FORMULATION

A. System Description

As an important part of sensing system in autonomous vehicles, trajectory prediction system is of great significance for understanding the surrounding environment and improving driving safety. At the same time, the effective prediction of the surrounding vehicle trajectory is also an important reference for the ego vehicle path planning. In this paper, we proposed a trajectory prediction method based on path planning, which considers collision risk at the same time.

The system is mainly composed of four parts, as shown in Figure 1. The vehicles in the figure are divided into three categories, the target vehicle is defined as ego vehicle and represented in white. Then the vehicles with X values greater than ego vehicle are defined as front vehicles and represented by blue vehicles, on the contrary, the vehicles with X values less than ego vehicle are defined as rear vehicles and represented by red vehicles. The first part of the system is to get the planning goals. According to Y. Hu et al [3], we use the rear-drivable areas of the front vehicles as the planning goals. The second part is the future trajectories prediction of the rear vehicles, then the trajectories are predicted by the constant velocity model [7] and as an important basis for collision assessment. The third part is the core part of the system, according to the planning goals and the risk of collision with rear vehicles, we get a series of non collision and reasonable planning trajectory of ego vehicle. Then we choose the best one according to the last two frame state of the ego vehicle.

B. Problem Statement

We assume that the position (x_i^t, y_i^t) of vehicle i at time-step t is represented by p_i^t . The goal of vehicle motion prediction is to predict the future trajectory, for the target vehicle i , we observe its position $H_i = (p_i^0, \dots, p_i^{t_{obs}})$ from time 1 to t_{obs} , and predict its future trajectory $T_i = (p_i^{t_{obs}+1}, \dots, p_i^{t_{pred}})$ for time instants $t_{obs}+1$ to t_{pred} , taking into account the state of

surrounding vehicles and collision risk with rear vehicles. The planning-based method that estimates the future trajectory T_i can be formulated as

$$\arg \min_{\theta} P_{\theta} \left(T_i^j | T_i^0, \dots, T_i^n \right), \quad (1)$$

where θ are the model's parameters and n the number of planning goals in the scene. This problem is often viewed as a sequence generation problem [12], where the model only considers the history trajectory.

III. METHOD

In this paper, we use one of the planning-based approaches called forward planning approaches, which uses optimal motion and path planning techniques with a hand-crafted cost-function, the overall structure of the method is shown in the Figure 2.

Algorithm 1 planning-based prediction algorithm

Input: *history_trajectory*
Output: *predicted_path*

- 1: *search for around vehicles as obstacles*
- 2: *estimate around vehicles' trajectories as linear motion*
- 3: $s_{start} \leftarrow \text{history_trajectory}[-1]$
- 4: $g(s_{start}) \leftarrow 0$
- 5: $g(\text{the rest of the states}) \leftarrow \infty$
- 6: $OPEN \leftarrow \{s_{start}\}$
- 7: $CLOSED \leftarrow \emptyset$
- 8: *PreprocessMap()*
- 9: **while** *s_{goal} is not expanded do*
- 10: *removes with the smallest f(s) from OPEN*
- 11: **for** each successors' of *s do*
- 12: **if** $g(s') > g(s) + c(s, s')$ **then**
- 13: $g(s') \leftarrow g(s) + c(s, s')$
- 14: **end if**
- 15: *lookup/update heuristic_function_memory*
- 16: *insert/update OPEN with $f(s') \leftarrow g(s') + \epsilon * h_{goal}(s') + h_{obst}(s')$*
- 17: **end for**
- 18: **end while**
- 19: *predicted_path* \leftarrow *path with smallest f(s) in CLOSED*
- 20: **return** *predicted_path*

A. Planning Goals Constraints

According to Figure 2, we firstly need to define the planning goals before realizing the planning algorithm. The dataset we use in this paper is the highD dataset, which is described in section IV. The average speed in the dataset is 40 km/h, and the Distance Headway (DHW) parameter is also provided. The value is set to 0, if no preceding vehicle exists. Considering the average speed and the DHW value in the scenario of the dataset, we select a safe distance $d_{dis} = 30m$. For the current frame, if the x value of the surrounding vehicle is greater than the value of the ego vehicle, it will be taken as the reference point, and the area beyond the safe distance d_{dis} is regarded as the drivable area $\chi \in [x_i, x_j - d_{dis}] (j \neq i)$. In

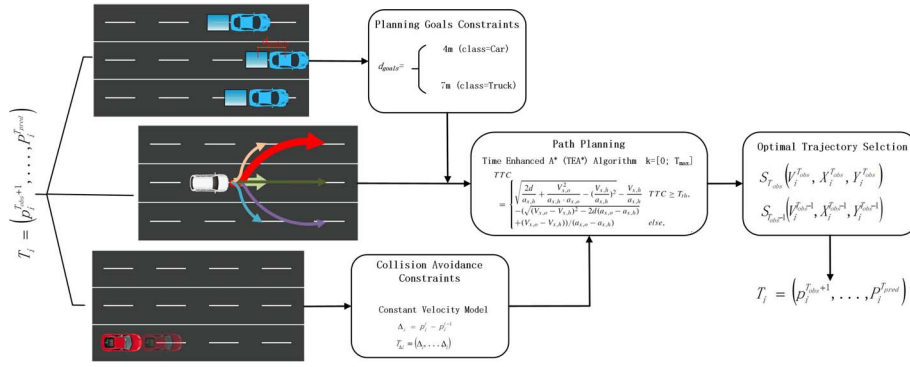


Fig. 2. The overall structure of the proposed trajectory prediction based on planning method.

each drivable area, $m = (x_j - d_{dis} - x_i)/d_{vehicle}$ target points are selected as the planning goals.

B. Path Planning Method

Due to the time dimension information involved in this task, the general two-dimensional planning algorithm is no longer suitable, so this paper uses an improved A* algorithm called time enhanced A* (TEA*) [18]. It is based on the Search Method A*. In our method, time is added to the traditional A* as the third dimension. The input map consists of three dimensions: the ego-vehicle coordinates (x_i and y_i) and time $t_i = [t_{obs+1}, \dots, T_{pred}]$. As shown in Figure 3, $k = [0; T_{max}]$ (T_{max} is the maximum number of layers) represents different time dimensions. TEA* calculates the minimum path over the temporal layers, considering the vehicles' positions and the changes of the environment.

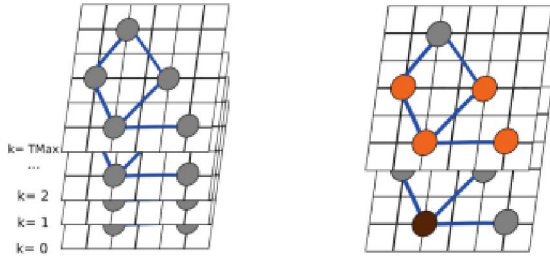


Fig. 3. The overall structure of the proposed trajectory prediction based on planning method.

In addition, the time dimension is being considered, there are some modifications compared with the traditional A*. Here are two definitions that are extremely important for TEA*.

- **Definition1:** The neighbor vertices of a vertex j in the temporal layer k belongs to the next temporal layer given by $k + 1$ (Figure 3)
- **Definition2:** The neighbor vertices of vertex j (v_{adj}^j) include the vertex containing the vehicle current position, and all adjacent vertices in the next time component.

The description of the planning-based prediction algorithm is presented in Algorithm 1, $g(s')$ represents the total cost function and can be expressed by Eq. 2. $c(s, s')$ represents

the single-step cost function, which can be expressed by Eq. 3. Eq. 5 $h_{goal}(s')$ represents the target heuristic function. Eq. 6 $h_{obs}(s')$ represents the obstacle safety heuristic function.

$$g(s') = g(s) + c(s, s') \quad (2)$$

$$c(s, s') = \sqrt{(x_{s'} - x_s)^2 + (y_{s'} - y_s)^2 + (z_{s'} - z_s)^2} \quad (3)$$

$$f(s') = g(s') + \epsilon_{goal} * h_{goal}(s') + \epsilon_{safe} * h_{obs}(s') \quad (4)$$

$$h_{goal}(s') = \sqrt{(x_{s'} - x_{goal})^2 + (y_{s'} - y_{goal})^2} \quad (5)$$

$$h_{obs}(s') = \min(d_{obs1,s'}, d_{obs2,s'}, d_{obs3,s'}, \dots) \quad (6)$$

C. Collision Avoidance Constraints

In the process of determining the planning goals, the minimum x_{min} in the drivable area can be consistent with the x_i of the ego vehicle. Obviously, this situation is not satisfied with the vehicle dynamics, and the rear vehicle may also collide with it. Therefore, we use the vehicle longitudinal collision avoidance model for collision risk assessment. As we all know, time to collision (TTC) is the most common metrics to calculate the collision risk, it can be calculated as shown in Eq. (7) and (8) [19]. d is the distance between the rear vehicle and obstacle vehicle (ego vehicle planning path), $V_{x,r}$ and $a_{x,r}$ represent the longitudinal velocity and acceleration of the rear vehicle, $V_{x,o}$ and $a_{x,o}$ represent the longitudinal velocity and acceleration of the obstacle vehicle. Based on the calculated TTC, the trajectory sets obtained in the previous planning are preliminarily selected.

$$TTC = \begin{cases} \sqrt{\frac{2d}{a_{x,r}} + \frac{V_{x,o}^2}{a_{x,r}a_{x,o}} - \left(\frac{V_{x,r}}{a_{x,r}}\right)^2} - \frac{V_{x,r}}{a_{x,r}} & TTC \geq T_{th} \\ -\left(\sqrt{(V_{x,o} - V_{x,r})^2 - 2d(a_{x,o} - a_{x,r})} + (V_{x,o} - V_{x,r})\right) / (a_{x,o} - a_{x,r}) & else \end{cases} \quad (7)$$

$$T_{th} = -\frac{V_{x,o}}{a_{x,o}} \quad (8)$$

D. Optimal path Selection

According to the research of Schöller et al. [7], it is possible to make a better prediction of vehicle behavior by observing the current state of the target vehicle. In this approach, we use the state information of the current frame (t_{obs}) and the previous frame ($t_{obs} - 1$) to select the best planning trajectory, including speed (v_i^{obs}, v_i^{obs-1}) and position deviation ($x_i^{obs} - x_i^{obs-1}, y_i^{obs} - y_i^{obs-1}$). Through the speed deviation ($\Delta V = |v_i^{obs} - v_i^{obs-1}|$) of the two frames, it can be determined whether the vehicle keeps straight or has the intention of steering. In addition, the turning left or the turning right can be estimated by the lateral deviation ($\Delta Y = y_i^{obs} - y_i^{obs-1}$). With these clues, we can find the most suitable planning path.

IV. EXPERIMENTS

A. Datasets

The highD dataset [13] is a new dataset of naturalistic vehicle trajectories recorded on German highways by a drone. In total, it is recorded at six different locations and different traffic states, it is also a large-scale dataset that includes more than 110,500 vehicles, 44,500 driven kilometers and 147 driven hours of vehicle trajectory data at 25 Hz. In addition, the track of each vehicle includes vehicle type, size, driven maneuvers and metrics like THW.

In highD dataset, the origin of the coordinate system is in the upper of left corner, as shown in the Figure 4. The horizontal axis is the x-axis which grows to the right, the vertical axis is the y-axis and it grows downwards.



Fig. 4. The coordinate system of highD dataset

In our experiments, we follow the experimental protocol of [7], where randomly split the training scenes into a training set and a 10% validation set to detect overfitting. We extract 8 seconds trajectories, using the first 3.2 seconds as an observation windows and the next 4.8 seconds as a prediction, which is an established setting and used in other motion prediction papers as well [14], [15], [16], [17].

B. Experiment Design

Based on Algorithm 1, we implement the model in Python 3.6 software. Experiments were executed on an E5-2678 CPU with 16 GB RAM that runs at 2.50×2 GHz. In order to correspond with other motion prediction papers, we observe the last 3.2 seconds of the ego vehicle trajectory and predict the next 4.8 seconds. Because the frequency of the dataset is

25 Hz, it is equivalent to observing 80 frames and predicting 120 frames. The total length of the trajectory is 200 frames, which is directly discarded if the length is less than 100 frames.

In order to evaluate our planning-based prediction method, we trained a Long Short-term Memory encoder-decoder (LSTM-ED) [20] and a Constant Velocity Model (CVM) [7] as the baselines.

- *LSTM-ED* — Encoder network that receives the input sequence $u_1, \dots, u_{t_{obs}}$ of the length t_{obs} and produces the summary of the past input sequence through the cell state vector c_t . Decoder network recursively generates the output sequence $s_1, \dots, s_{t_{pred}}$ of the length t_{pred} .
- *CVM* — It assumes that the ego vehicle will continue to motion with the same velocity and direction as observed from the latest two timesteps. The future trajectory as $T_{\Delta i} = (\Delta i, \dots, \Delta i)$, the number of Δi is equal to the number of prediction steps.

Then we use common metrics as the evaluation criteria for the experimental results, and the unit is meter.

- *AverageDisplacementError* (ADE) — Average L2 distance between position of the predicted trajectory and the ground truth that have the same temporal distance from their respective start position.
- *FinalDisplacementError* (FDE) — L2 distance between final predicted position and the ground truth position at the corresponding time point.

C. Results and Analysis

Table I shows the prediction errors for all models on each scene. We first compare the ADE values of prediction error between the models, the CVM has the best performance, and our model has made great progress compared with LSTM-ED. The dataset used in our experiment is a highway scene and the selected observation area is a straight road. In addition, the surrounding environment is simple, and the vehicles usually go straight and seldom perform other maneuvers, which is very beneficial to the CVM. On the other hand, our method first selects a drivable area to filter out some unreasonable points, so the accuracy is higher than LSTM-ED. Then, we investigate how these models behaves with respect to the metric FDE. It has the same trend as the previous ADE, but our model has planning goals which ensure the planning path in the drivable area. As a result, there is not a large deviation error compared to LSTM.

TABLE I
DISPLACEMENT ERRORS OF THE BASELINES AND OURS

Metric	CVM	LSTM-ED	OURS
ADE	0.71	4.13	2.28
FDE	1.77	5.36	3.25

Based on the above analysis, we need to further verify the performance of our model on more complex dataset (such as inD dataset [21], INTERACTION Dataset [22]) in the future.

V. CONCLUSIONS AND FUTURE RESEARCH

In this paper, a novel planning-based method for trajectory prediction was proposed. The principle of the system and model design are described in detail. By calculating TTC, the collision risk is considered. Then we have compared our method against two baselines in highD dataset, our model's deviation error accuracy is higher than the LSTM-ED, but it is worse than CVM. Because of the characteristics of highD dataset, it has a positive effect on CVM. Therefore, we plan to do some experiments on more complex dataset in the future to verify that our model is more comprehensively.

ACKNOWLEDGMENT

This work was supported by the Youth Program of National Natural Science Foundation of China (Grant No. 61906138) and Shanghai AI Innovation Development Program 2018.

REFERENCES

- [1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33-55, Mar. 2016.
- [2] S. Hoermann, D. Stumper and K. Dietmayer, "Probabilistic long-term prediction for autonomous vehicles," *2017 IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, 2017, pp. 237-243.
- [3] Y. Hu, W. Zhan and M. Tomizuka, "Probabilistic prediction of vehicle semantic intention and motion," *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, 2018, pp. 307-313.
- [4] H. Cui et al., "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 2090-2096.
- [5] Nachiket Deo, Mohan M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018, pp. 1468-1476.
- [6] D. Vasquez, "Novel planning-based algorithms for human motion prediction," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, 2016, pp. 3317-3322.
- [7] C. Schöller, V. Aravantinos, F. Lay and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696-1703, Apr. 2020.
- [8] Rudenko, Andrey, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrilă, and Kai O Arras. "Human motion trajectory prediction: a survey," *The International Journal of Robotics Research*, pp. 0278364920917446, Jun. 2020.
- [9] D. Vasquez, "Novel planning-based algorithms for human motion prediction," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, 2016, pp. 3317-3322.
- [10] F. Previtali, A. Bordallo, L. Iocchi and S. Ramamoorthy, "Predicting future agent motions for dynamic environments," *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Anaheim, CA, 2016, pp. 94-99.
- [11] A. Houénou, P. Bonnifait and V. Cherfaoui, "Risk assessment for collision avoidance systems," *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, 2014, pp. 386-391.
- [12] Graves, Alex, "Generating sequences with recurrent neural networks," In *arXiv preprint arXiv:1308.0850*, 2013.
- [13] R. Krajewski, J. Bock, L. Kloeker and L. Eckstein, "The highd dataset: a drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, 2018, pp. 2118-2125.
- [14] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, Nanning Zheng, "Sr-lstm: state refinement for lstm towards pedestrian trajectory prediction," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12085-12094.
- [15] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezafofghi, Silvio Savarese, "Sophie: an attentive gan for predicting paths compliant to social and physical constraints," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1349-1358.
- [16] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, Alexandre Alahi, "Social gan: socially acceptable trajectories with generative adversarial networks," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2255-2264.
- [17] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: human trajectory prediction in crowded spaces," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961-971.
- [18] J. Santos, P. Costa, L. F. Rocha, A. P. Moreira and G. Veiga, "Time enhanced A*: towards the development of a new approach for multi-robot coordination," *2015 IEEE International Conference on Industrial Technology (ICIT)*, Seville, 2015, pp. 3314-3319.
- [19] E. Coelingh, A. Eidehall and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection - a practical example of automatic emergency braking," *13th International IEEE Conference on Intelligent Transportation Systems*, Funchal, 2010, pp. 155-160.
- [20] S. H. Park, B. Kim, C. M. Kang, C. C. Chung and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, 2018, pp. 1672-1678.
- [21] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: a drone dataset of naturalistic road user trajectories at german intersections," In *arXiv preprint arXiv:1911.07602*, 2019.
- [22] Zhan, W., Sun, L., Wang, D., Shi, H., Clause, A., Naumann, M., ... Tomizuka, M., "Interaction dataset: an international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," In *arXiv preprint arXiv:1910.03088*, 2019.