ReCoAt: A Deep Learning Framework with Attention Mechanism for Multi-Modal Motion Prediction

Zhiyu Huang, Xiaoyu Mo, Chen Lv
Nanyang Technological University
zhiyu001@e.ntu.edu.sg, xiaoyu006@e.ntu.edu.sg, lyuchen@ntu.edu.sg

Abstract

This work proposes a novel deep learning framework for multi-modal motion prediction. The framework consists of three parts: recurrent neural network to process target agent's motion process, convolutional neural network to process the rasterized environment representation, and distance-based attention mechanism to process the interactions among different agents. The qualitative results manifest that the predicted trajectories given by our model are accurate, diverse, and in accordance with the road structure. The quantitative results of the metrics on the standard test set reveal that our model outperforms other baseline methods in terms of prediction accuracy.

1. Introduction

Motion prediction is one of the most crucial and challenging tasks for autonomous driving. Accurately predicting traffic participants' future trajectories is essential for autonomous vehicles to make informed and human-like decisions and plan safe and efficient motion trajectories, especially in crowded and complex traffic scenarios. However, it is a remarkably challenging task due to the following factors. First and foremost, in addition to the dynamics or physical constraints, the future trajectories of the traffic participants are highly dependent on the environment information. This encompasses the information of the map (e.g., drivable lanes for vehicles and crossing for pedestrians), traffic signals (e.g., traffic lights and stop signs), traffic rules, and more importantly the interactions among different traffic participants. Another influential factor is the uncertainties in intention and behavior. There may be a variety of plausible future motion trajectories for the target due to unknown destinations and noisy movements. This requires the motion prediction model to be able to output multiple possible trajectories and their likelihoods.

In this work, a deep learning framework is proposed to output multi-modal motion trajectories by modeling the interaction between different agents and exploiting the environmental context information, as well as the dynamics information of the target agent. As for modeling the interaction between traffic participants, many recent works attempt to use the attention mechanism. For example, [6] utilize multi-head self-attention to account for interactions among different vehicles, and [2] and [4] propose to utilize graph attention networks to extract relational features on the scene graph containing different agents. Considering that in a traffic scenario, the attention the target agent pays to a surrounding agent is largely determined by the distance between them, we propose a distance attention module to model the interaction between the target agent and its surrounding agents.

2. Framework

2.1. Problem formulation

The task of motion prediction is to predict the possible future trajectories of a target agent over a time horizon T_f based on its historical states over a time period T_h and environmental context information. The input $\mathbf{X}$ to the prediction model consists of the historical dynamic states of itself (S_0) and its surrounding agents ($S_1, \dots, S_N$), as well as the current environment information $\mathcal{M}$. Without loss of generality, we assume that there are N surrounding agents around the target agent, however, the number of surrounding agents can be varied in different situations. The agents include vehicles, pedestrians, and cyclists. The output of the prediction model $\hat{\mathbf{Y}}$ is K trajectories, each consisting of a sequence of 2D coordinates denoting the possible future positions of the target agent. Mathematically, the problem is formulated as:

$$\begin{aligned} \mathbf{X} &= \{S_0, S_1, \dots, S_N, \mathcal{M}\}, \\ \hat{\mathbf{Y}} &= \{(x_j^t, y_j^t) | t \in \{t_o + 1, \dots, t_o + T_f\}\}_{j=1}^K, \end{aligned} \tag{1}$$

where $S_i = \{s_i^{t_o - T_h + 1}, s_i^{t_o - T_h + 2}, \dots, s_i^{t_o}\}$ and s_i^t is the dynamic state of the agent i at timestep t, (x_j^t, y_j^t) is the jth predicted coordinate of the target agent at timestep t, and t_o is the current time step.

2.2. Dataset and data processing

We employ the Waymo open motion dataset [3], which is a large-scale and diverse motion forecasting dataset that contains over 100,000 driving scenes with interesting interactions between vehicles, pedestrians, and cyclists. The dataset gives agents' tracks for the past 1 second at 10Hz sampling and a corresponding map, and the motion prediction task is to predict the future positions of target agents for 8 seconds with a sampling rate of 2Hz.

The historical dynamic state of the target agent and its surrounding agents s_i^t is in the format of (x, y, v_x, v_y, θ), where (x, y) is the coordinate, (v_x, v_y) the velocity, and θ the heading angle. Note that the coordinate system is centered on the target agent's position at the current timestep with its heading aligned with the x-axis. Thus, for each agent, its state representation is a tensor with shape $(10, 5)$. We only consider up to ten surrounding agents within a radius of 30 meters to the target agents and incorporate the states of them into a fixed-length tensor with shape $(10, 10, 5)$. The surrounding agents in the tensor are ordered according to their distances to the target agent and the tensor is padded with zeros if there are not enough surrounding vehicles found.

The environmental information is represented as bird-eye-view rasterized 2D images, as shown in Fig. 1. The target agent represented as a red box is positioned at $(1/5, 1/2)$ of the image. The surrounding agents are shown in different colors: magenta for vehicles, blue for pedestrians, and green for cyclists. The tails (thin lines attached to the boxes) behind the agents are the historical tracks. The drivable lanes are displayed as grey polygons and the candidate centerlines for the target agent are painted in cyan. The candidate centerlines in the format of polylines (sequence of waypoints) are also used as additional information for predicting the vehicle's future motion. The red circles or green circles on the road show the states of the traffic lights and the red circles on the roadside represent the stop signs. The blue polygon is the pedestrian crossing and the orange polygon is the speed bump. The lane markings are represented in different kinds of polylines: yellow solid lines for road edges, white solid lines for solid white road lines, white dashed lines for broken white road lines, light yellow lines for yellow road lines. The bird-eye-view rasterized images are in the size of $240 \times 240 \times 3$ encoding different sizes of areas for different types of target agents. For vehicles, the actual area in the scene is $80m \times 80m$, and $60m \times 60m$ for cyclists and $40m \times 40m$ for pedestrians.

2.3. Model: ReCoAt

The structure of the motion prediction model named ReCoAt is visualized in Fig. 2. The name comes from different parts of the model, which are recurrent neural network (RNN) to process the target agent's dynamic state,

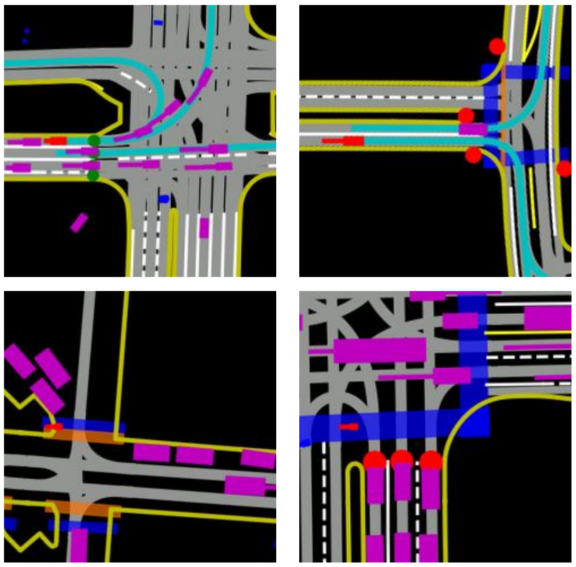


Figure 1. Examples of rasterized environment representation

convolutional neural network (CNN) to process environmental context, and attention module to process the interactions among agents. The environment representation in the format of bird-eye-view rasterized images is processed by a ResNet50 encoder [5], which outputs a 2048-dimension feature vector, followed by a fully connected layer to reduce its dimension to 128. The past trajectory of the target agent is processed by a trajectory encoder, which consists of a 1D convolutional layer and an LSTM layer, to extract the motion information of the target agent. The output of the trajectory encoder is a 128-dimension feature vector. To model the interaction between the target agent and its surrounding agents, the proposed distance attention module is employed and illustrated below.

First of all, all the surrounding agents' trajectories are projected to high-dimensional feature vectors by the trajectory encoder with shared weights across them. In accordance with the definition of the attention mechanism, the query is the current position of the target agent, keys are the positions of the surrounding agents, values are the feature vectors obtained by the trajectory encoder. The score function, which rates which agents the target agent needs to pay attention to, is defined as:

$$f_{att}(key_i, query) = \frac{\alpha}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}, \quad (2)$$

where α is a hyper parameter, (x_0, y_0) is the position of the target agent, and (x_i, y_i) is the position of the surrounding agent i. The intuition behind this score is that the target agent needs to pay more attention to agents that are closer to it and less attention to those that are still far away. The attention weights are calculated by a softmax over all score

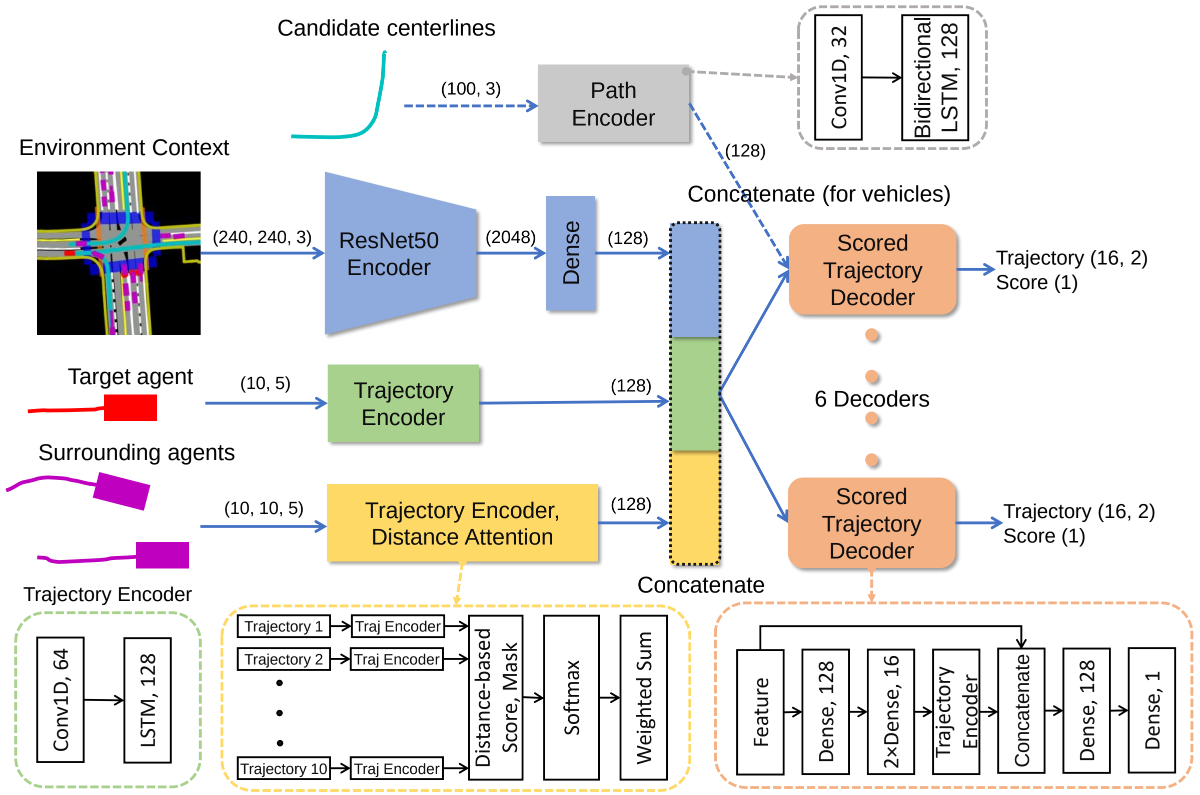


Figure 2. The structure of the ReCoAt motion prediction model

function outputs:

$$\beta_i = \frac{\exp f_{att}(key_i, query)}{\sum_j \exp\left(f_{att}(key_j, query)\right)}. \tag{3}$$

The output of the attention module is calculated as the weighted sum of the value vectors:

$$out = \sum_i \beta_i \cdot value_i. \tag{4}$$

In practice, we need to mask out the padding elements by assigning the corresponding score function outputs with a large negative value.

Concatenating the feature vectors from the target agent, environment context, and agent interaction, we obtain a low-dimensional feature vector. For vehicles, we also add the information of candidate centerlines processed by a path encoder to the feature vector. We use a mixture of different trajectory decoders to output multi-modal trajectories and their associated scores. The trajectory is given by two dense layers representing x and y coordinates respectively. The scoring of the trajectory is conditioned on the feature vector and the encoded trajectory feature. The scores of all the predicted trajectories are then stacked and passed through a softmax layer to output the probabilities of these trajectories. The diversity among the trajectory decoders is encouraged by only updating the trajectory regression part of the winning decoder, of which the output trajectory is the closest to the ground truth, while the scoring part for all the decoders can be updated. This could help assign each training example to a particular mixture and also each mode to specialize for a distinct class of behaviors (e.g., going straight and turning) [1].

2.4. Loss functions

To improve the model's performance in long-term prediction and low-speed prediction, we use the weighted mean absolute error whose weights scale with time t and the initial speed of the target agent v. For a data point, its trajectory loss is defined as:

$$\mathcal{L}_{traj} = \min_{j\in\{1,\dots,K\}} \frac{1}{T_f} \sum_t w_t^v \parallel (x_{gt}^t, y_{gt}^t) - (x_j^t, y_j^t) \parallel, \tag{5}$$

where (x_{gt}^t, y_{gt}^t) is the ground truth position at time step t. The per-step weight w_t^v is defined as follows:

$$w_t^v = w_t \cdot w^v, \ w_t = 0.5t, \ w^v = \max(1, 4 - 0.2v). \tag{6}$$

The scoring loss is the cross entropy loss between the ground truth probability distribution and the predicted distribution. For a data point, the scoring loss is defined as:

$$\mathcal{L}_{score} = \mathcal{L}_{CE}(p_{gt}, p), \tag{7}$$

where p is the predicted probability distribution and the ground truth distribution p_{gt} is defined according to the L2 distance between the trajectory endpoint $\mathbf{s}^{T_f}$ and ground truth endpoint $\mathbf{s}_{gt}^{T_f}$:

$$p_{gt} = \frac{\exp - \parallel \mathbf{s}^{T_f} - \mathbf{s}_{gt}^{T_f} \parallel_2}{\sum_j \exp - \parallel \mathbf{s}_j^{T_f} - \mathbf{s}_{gt}^{T_f} \parallel_2}. \tag{8}$$

The total loss is a weighted sum of the trajectory loss and the scoring loss:

$$\mathcal{L} = \mathcal{L}_{score} + \lambda \mathcal{L}_{traj}, \tag{9}$$

where λ is a hyperparameter to balance the training process.

2.5. Implementation details

All the activation functions in the dense layers are the ELU, and all the dense layers are followed by dropout layers to mitigate overfitting. The parameters of other layers follow the default settings in Tensorflow. The parameter α in the attention module is set as 10 and the parameter λ in the loss function is 0.2. Different types of target agents (vehicles, cyclists, and pedestrians) are grouped together to train a prediction model. We use Nadam optimizer with a learning rate that starts with 6e-4 and decays by 0.9 after every epoch. The number of training epochs is 50 and the batch size is 32.

3. Results

3.1. Qualitative results

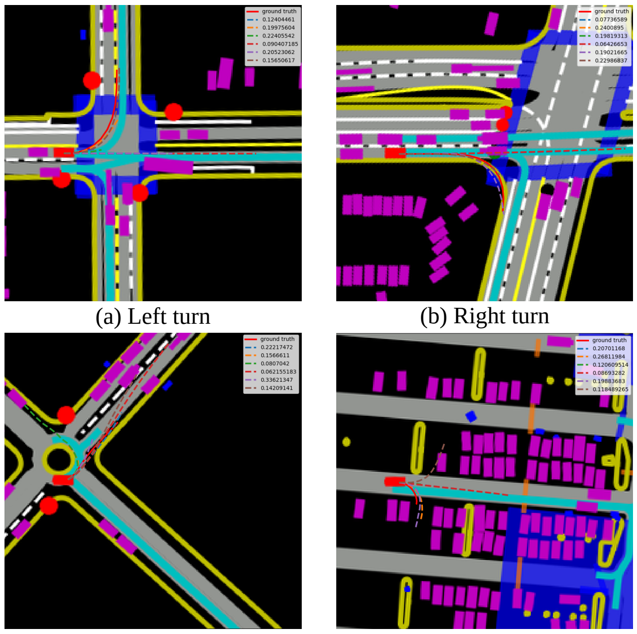
Fig. 3 shows some examples of multi-modal motion prediction given by our model. The results reveal that our model can accurately predict multiple possible trajectories and their probabilities in different traffic scenarios. The predicted trajectories are diverse and in accordance with the road structure, and those trajectories that are close to the ground truth are scored higher.

3.2. Quantitative results

Table 1 gives the results of average metrics of our model on the standard test set of the Waymo motion dataset, in compassion with the LSTM baseline method. The metrics are averaged over different object types and different evaluation times (3, 5, and 8 seconds). The results reveal that our model shows improved prediction accuracy in both trajectory error and scoring compared to the baseline method.

Method	minADE (m) ↓	minFDE (m) ↓	Miss Rate ↓	Overlap Rate ↓	mAP ↑
LSTM	1.0065	2.3553	0.3750	0.1898	0.1756
Ours	**0.7703**	**1.6668**	**0.2437**	**0.1642**	**0.2711**

Table 1. Average metrics on the standard test set of the Waymo motion dataset



(a) Left turn (b) Right turn

(c) Roundabout (d) Parking lot

Figure 3. Examples of the model predictions

References

[1] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019. 3

[2] Bo Dong, Hao Liu, Yu Bai, Jinbiao Lin, Zhuoran Xu, Xinyu Xu, and Qi Kong. Multi-modal trajectory prediction for autonomous driving with semantic map and dynamic graph attention network. *arXiv preprint arXiv:2103.16273*, 2021. 1

[3] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. *arXiv preprint arXiv:2104.10133*, 2021. 2

[4] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 2

[6] Jean Mercat, Thomas Gilles, Nicole El Zoghby, Guillaume Sandou, Dominique Beauvois, and Guillermo Pita Gil. Multi-head attention for multi-modal joint vehicle motion forecasting. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9638–9644. IEEE, 2020. 1