Social relation and physical lane aggregator: integrating social and physical features for multimodal motion prediction

Qiyuan Chen, Zebing Wei and Xiao Wang
The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China and School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
Lingxi Li
Electrical and Computer Engineering, Indiana University-Purdue University Indianapolis, Indianapolis, Indiana, USA, and
Yisheng Lv
Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China and School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

Abstract

Purpose – The purpose of this paper aims to model interaction relationship of traffic agents for motion prediction, which is critical for autonomous driving. It is obvious that traffic agents' trajectories are influenced by physical lane rules and agents' social interactions.

Design/methodology/approach – In this paper, the authors propose the social relation and physical lane aggregator for multimodal motion prediction, where the social relations of agents are mainly captured with graph convolutional networks and self-attention mechanism and then fused with the physical lane via the self-attention mechanism.

Findings – The proposed methods are evaluated on the Waymo Open Motion Dataset, and the results show the effectiveness of the proposed two feature aggregation modules for trajectory prediction.

Originality/value – This paper proposes a new design method to extract traffic interactions, and the attention mechanism is used in each part of the model to extract and fuse different relational features, which is different from other methods and improves the accuracy of the LSTM-based trajectory prediction method.

Keywords Deep learning, Machine learning, Autonomous driving, Trajectory prediction

Paper type Research paper

1. Introduction

Motion prediction is a crucial component of developing autonomous vehicles (Wang, 2010). However, it is a hard problem due to the complexity of traffic agents' dynamical moving and social interaction process. To better predict traffic agents' good motion behavior, it is necessary to consider social interactions between traffic agents and the physical constraints of the roads in a given scene.

Traffic prediction based on deep learning has been developed for a long time (Lv *et al.*, 2014; Wang *et al.*, 2017; Chen *et al.*, 2022; Zhang *et al.*, 2021; Li *et al.*, 2021; Wei *et al.*, 2021). There have been many methods for motion prediction of pedestrians and vehicles. How to model the target traffic agent's interactions with nearby traffic agents and the road environment is a core issue. Social interactions between traffic agents mainly focus on interactions between the target agent and its surrounding vehicles, pedestrians and nonmotorized vehicles, and it is a dynamic interaction relationship usually resulting from collision avoidance. There have been many approaches to tackle this task. Existing interaction modeling works can be classified into three categories: geometry-based modelling, image-based modelling and vector-based modelling. The former method takes the physical relationships between agents into account and models the relationships for agents artificially. The delineation of such relationships is often based on real-life knowledge or common sense and is modeled

The current issue and full text archive of this journal is available on Emerald Insight at: **https://www.emerald.com/insight/2399-9802.htm**

Journal of Intelligent and Connected Vehicles
5/3 (2022) 302–308
Emerald Publishing Limited [ISSN 2399-9802]
[DOI 10.1108/JICV-07-2022-0028]

This work was supported by Chinese Guangdong's S&T project 2019B1515120030 and National Natural Science Foundation of China under Grant 61876011.

Received 9 July 2022
Revised 29 July 2022
Accepted 29 July 2022

from a specific perspective, such as relative distance or speed. Alahi *et al.* (2016) performed a social tensor which was known as the social pooling layer, to represent geometric relationships between agents. A similar approach using geometric relationships to help construct the potential interaction features for pedestrians has been applied to vehicle motion prediction. Deo and Trivedi (2018) used convolutional social pooling as an improvement to social pooling layers for robustly learning interdependencies in vehicle motion. Convolutional social pooling can provide good insight into the relative position information between agents but cannot cover the changes in their relative position. Shi *et al.* (2021) used Graph Convolutional Networks (GCN) to extract interactions between pedestrians, which was also a method to extract relations based on geometry.

The image-based works assign different elements of the scene to channels and then overlay this information on a single image. Konev *et al.* (2022) divided images into 25 channels, the first three are Red, Green, Blue channels containing road information such as lane lines and traffic signals, the 4th to 14th channels are the positions of the target agent at each moment, and the positions of others at each timestep are described in the last 11 channels, then Convolutional Neural Networks (CNN) backbone pretrained on ImageNet is used to predict. Although it is believed that tensor and convolution can learn better spatiotemporal interactions among agents and environments, the data needs to be processed into images in advance, which is complicated and time-consuming. At the same time, the proportion of images containing features is small, and those blank areas can also affect computing ability.

The trajectories of traffic agents are likewise influenced by the physical lanes. The road polylines are represented as a collection of vectors to aggregate the information on lanes, and each polyline is aggregated into a vector representation in the work of Gao *et al.* (2020). That is also how Gu *et al.* (2021) and Zhao *et al.* (2020) did. Another popular method of physical lane information aggregation is to rasterize lanes and trajectories into a graph, from which learns the relationship between agents and lanes (Konev *et al.*, 2022). Figure 1 shows several methods of feature aggregation. In Figure 1(a), the interaction representation of agents is based on geometric relations (Song *et al.*, 2020), then Figure 1(b) shows the rasterized maps in which different channels contain different information, and this approach often uses convolutional neural networks to aggregate features and Figure 1(c) shows the map generated by VectorNet.

In this paper, we propose a social relation and physical lane aggregator, which is a new structure to aggregate social interaction features of traffic agents and physical lane features for motion prediction. The proposed model takes lane point coordinates and vehicle history track points as input, as shown in Figure 2, and the overall model can be divided into three parts: social aggregator, hybrid interaction aggregator and the decoder. For the social relation aggregator, self-attention and GCN are used to aggregate the features of traffic agents in a given scene; and for the hybrid information of social relation and physical lane, VectorNet and self-attention are both used to aggregate features between lane and agents jointly.

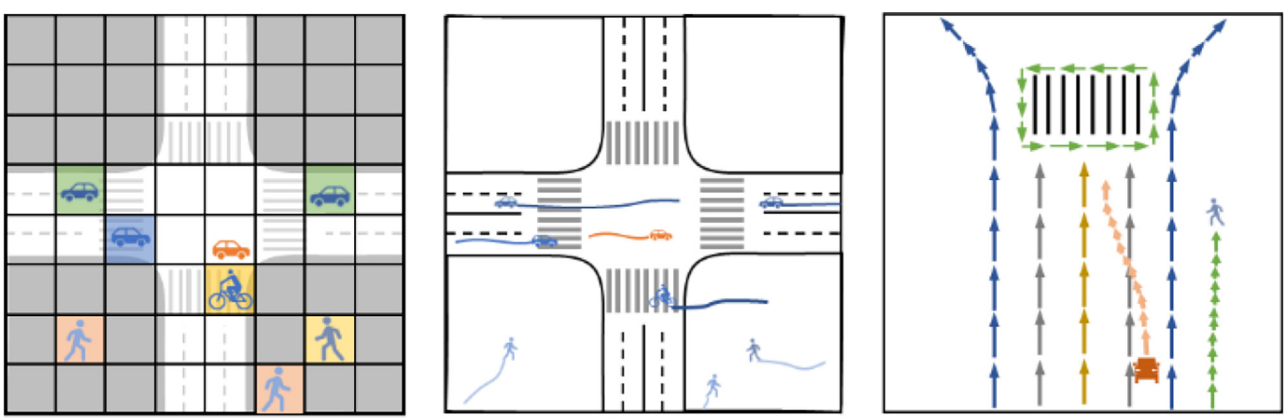
Then the aggregated features are fed into the decoder, which is composed of Long short-term memory (LSTM) and Multilayer Perceptron (MLP) layers, to predict future trajectories and their confidence. Comprehensive experiments are conducted on Waymo Open Dataset to demonstrate the effectiveness of the proposed approach. In comparison with the LSTM-based method, our method achieves better results.

The rest of this paper is organized as follows. Section 2 gives the proposed social relation and physical lane aggregator. Section 3 shows the experimental results. Section 4 draws conclusions.

2. Methodology

Motion prediction aims to predict future location coordinates of traffic agents in a given scene. Given a series of observed trajectories over time $t \in \{1,2,\ldots,T_{Obs}\}$, the coordinates of traffic agent i at each timestep $\{p_t^i \mid (x_t^i, y_t^i), \quad t = 1, 2, \ldots, T_{obs}\}$, and N coordinates lane points in the scene $\{l^n \mid (x^n, y^n), n \in N\}$,

Figure 1 Several methods of feature aggregation



(a) (b) (c)

Figure 2 Model overview

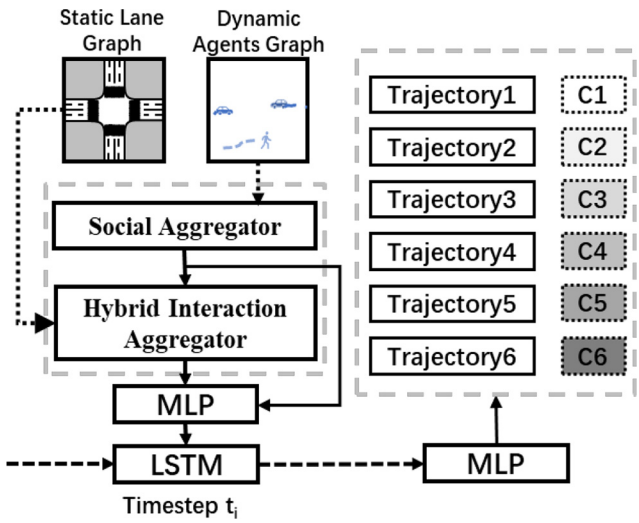
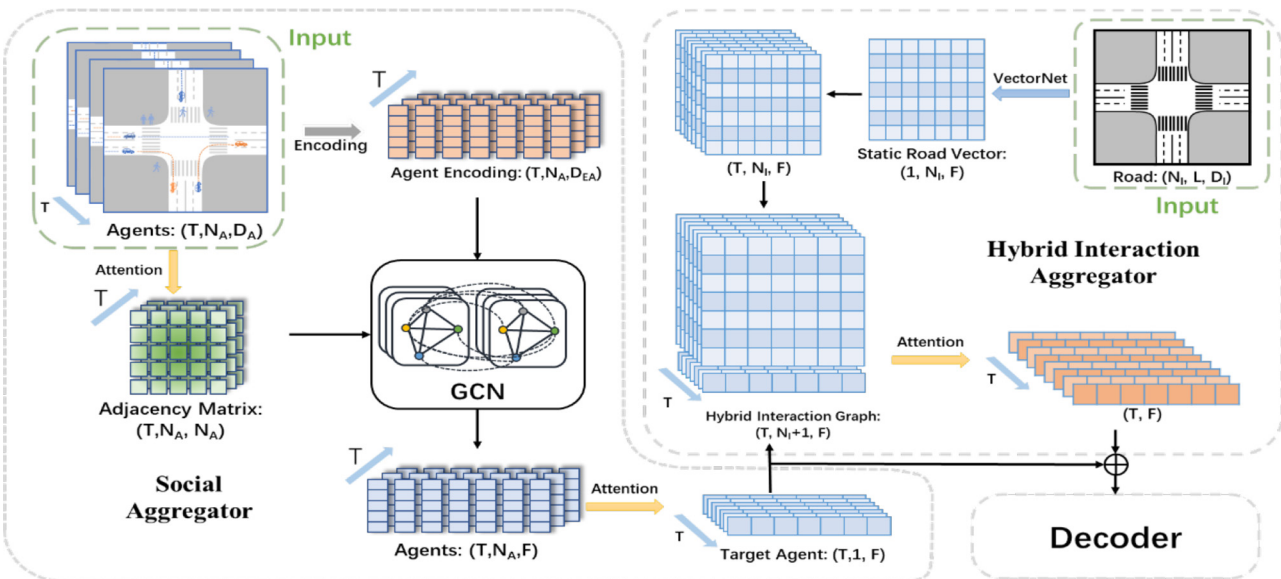


Figure 3 Details of social aggregator and lane aggregator



the goal is to predict its positions $\{p_{t'}^i \mid \left(x_{t'}^i, y_{t'}^i\right), t' = T_{obs}+1, T_{obs}+2, \ldots, T_{pred}\}$ in the future T_{pred} prediction horizon.

In this paper, social relation and physical lane aggregator are proposed to model both interactions between traffic agents and interactions between traffic agents and physical lanes, which is shown in Figure 3.

In the social relation aggregator module, the self-attention mechanism is used to encode agents' interaction features for context understanding. The attention matrix, which represents the relationships between agents, is fed into GCN as an adjacency matrix for interaction extraction. Similarly, the hybrid interaction aggregator is proposed to extract the relationship between traffic agents and physical lanes. The outputs of the two aggregators are fed together into a decoder composed of MLP and LSTM for trajectory prediction.

2.1 Social aggregator

2.1.1 Agents encoding

The input of the social aggregator is the coordinates, velocities, heading angles and agent types in a given scene. The target agents' location is set as the origin, and the observed data is normalized with respect to it. Each agent's feature G_a^i is given by:

$$G_a^i = \left(x_a^i, y_a^i, v_{ax}^i, v_{ay}^i, yaw^i, type_a^i\right) \tag{1}$$

where x_a^i and y_a^i are coordinates of agent i, v_{ax}^i and v_{ay}^i denotes the velocity along the axis, yaw^i and $type_a^i$ are heading angles and one-hot encoding of agent type. The encoding process is implemented through a two-layer MLP:

$$E_{spa} = MLP\left(G_a, W_E^{spa}\right) \tag{2}$$

where E_{spa} are encoded vectors, $W_E^{spa} \in D \times D_E^{spa}$ is the weights of MLP.

2.1.2 Interaction extractor

To capture the interaction between traffic agents, we first use the self-attention mechanism to compute the asymmetric adjacency matrix, i.e. the spatial interaction $A_{spa} \in R^{N\times N}$:

$$Q_{spa} = \varphi\left(E_{spa},\ W_Q^{spa}\right) \tag{3}$$

$$K_{spa} = \varphi\left(E_{spa},\ W_K^{spa}\right) \tag{4}$$

$$A_{spa} = Softmax\left(\frac{Q_{spa}K_{spa}^T}{\sqrt{d_{spa}}}\right) \tag{5}$$

where φ denotes linear transformation, Q_{spa} and K_{spa} are the query and the key, respectively. $W_Q^{spa} \in D \times D_Q^{spa}$, $W_K^{spa} \in D \times D_K^{spa}$ are weights of the linear transformations and $\sqrt{d_{spa}}$ is a scaling factor.

Because the adjacency matrix, A_{spa} is computed independently at each time step, and it does not contain any temporal information of the trajectory, which is the same as the distance-based adjacency matrix. But the difference is that, compared to using only distance as the adjacency matrix, the attention-based matrix has the advantage of covering more information, such as heading angles and velocity direction vectors. We stack A_{spa} from each time step as $A_{spa}^t \in R^{T_{obs}\times N\times N}$, and feed it into GCN for fusion. Each of the adjacency matrix is an asymmetric square matrix, where the (i, j) element of the matrix represents the influence of agent i to agent j; the relationship between them is asymmetrical. In the work of Shi *et al.* (2021), the adjacency matrix A_{spa}^t is convolved over row and column separately to predict for all agents, while we are making individual predictions and focusing only on initiative relations of the target, so only the rows are convolved:

$$E_{raw}^t = conv\left(E_{spa}^t, A_{spa}^t\right) \tag{6}$$

$$E_{gcn}^t = \sigma\left(E_{raw}^t\right) \tag{7}$$

where E_{spa}^t is the encoded vector at t time step, A_{spa}^t is the asymmetric adjacency matrix at that moment and σ is the activation function. In the traditional GCN, the adjacency matrix is fixed, but the interaction between agents changes at each time step, so convolution operation is performed at each time step. As shown in the social aggregator in Figure 3. The interaction feature algorithm is given in Algorithm 1.

Algorithm 1: Interaction Feature Fusion Algorithm
Input : Adjacency matrix A_{spa}, features of agents after encoding E_{spa};
Output: Features of *ith* interest after aggregation E_{agg}^i;
1: **for** $t = 0,1,2,\ldots,T_{pred}$ **do**
2: Calculate the convolution at the tth timestep and $E_{gcn}^t = \sigma\left(conv\left(E_{spa}^t, A_{spa}^t\right)\right)$;
3: **for** $t = 0,1,2,\ldots,T_{pred}$ **do**
4: Fusion by attention mechanisms $E_{att}^t = atten\left(E_{gcn}^t\right)$;
5: Output features of ith interest $E_{agg}^i = E_{att}[:, :, i, :]$.

2.2 Hybrid interaction aggregator

2.2.1 Physical lane extractor

The processing methods for road elements can be mainly classified into rasterization and vectorization. Combined with the output of social aggregator, it is more convenient to model road features using vectorization. The method of vectorizing road elements can capture the structural features of (High Definition) maps more efficiently.

VectorNet is a hierarchical graph neural network composed of a subgraph module and a global graph module (Gao *et al.*, 2020). The subgraph module is used to encode the features of the lanes and the agents, and the global graph module uses the attention mechanism to capture the interactions among the lanes and the agents. The lane information needs to be represented as vector E_l to be aggregated with agents, and the subgraph of VectorNet is performed to vectorize lane features, which is given by:

$$G_l = (x_l,\ y_l,\ dir_{xl},\ dir_{yl},\ type_l) \tag{8}$$

$$E_l = subgraph(G_l) \tag{9}$$

where x_l, y_l are the coordinates of road lanes, dir_{xl}, dir_{yl} are the direction vectors and $type_l$ is the type of lanes.

2.2.2 Hybrid interaction extractor

The lane graph is the same at each timestep in the same scene; therefore, it can be described as a static graph. To combine with agents' interaction graphs in the time dimension after contextual encoding of the lane lines, the static road graph can be replicated along time and connected to the agents' interaction graph at each moment. Thus, the hybrid interaction graph G_h is obtained, which contains the unfused features of the target agents and lanes. More specifically, we use the attention mechanism to extract the attentional relationship between lanes and the target agent and fuse their features.

$$Q_{hy} = \varphi\left(G_h,\ W_Q^{hy}\right) \tag{10}$$

$$K_{hy} = \varphi\left(G_h,\ W_K^{hy}\right) \tag{11}$$

$$V_{hy} = \varphi\left(G_h,\ W_V^{hy}\right) \tag{12}$$

$$Z_{hy} = Softmax\left(\frac{\sum_{i=1}^{n_l+1} Q_{hy}^a K_{hy}^i}{\sqrt{d_{hy}}}\right) V_{hy} \tag{13}$$

where n_l is the number of lanes, W_Q^{hy}, W_K^{hy} and W_V^{hy} are weights of the query, key and value's linear transformations, and the key of each element that participated in the fusion are multiplied with Q_{hy}^a, which is the query matrix of the target agent, to obtain the weighted value of each element's attention to the target agent. Z_{hy} is the hybrid matrix of agent-lane interaction.

2.3 Temporal prediction decoder

The prediction of the trajectory consists of two parts, the future trajectory points' coordinates and the confidence level of each trajectory. LSTM is used for trajectory prediction in this paper. For the confidence, we apply the MLP to generate the K confidence scores for each of the trajectory proposals.

$$G_f = LSTMG_h^i \tag{14}$$

$$P_{traj} = MLP(\ G_f, W_f) \tag{15}$$

where G_f is the trajectory prediction result of LSTM, P_{traj} is the confidence and W_f is the weight of MLP.

2.4 Objective function

Simple MSE loss does not allow probabilistic modeling of multiple possible outcomes, and it showed poor performance in our preliminary experiments. In this case, our network outputs the means of the Gaussians while we fix the covariance of every Gaussian to be the identity matrix I.

For the loss, we compute the negative log probability of the ground truth trajectory under the predicted mixture of Gaussians with the means equal to the predicted trajectories and the identity Matrix I as covariance:

$$\begin{aligned} L &= -log \sum_k c_k N(X^{gt}; \mu = X_k, \sigma = I) \\ &= -log \sum_k c_k \prod_{t=1}^{T_{pred}} N\left(x_t^{gt}; x_{k,t}, 1\right) N\left(y_t^{gt}; y_{k,t}, 1\right) \\ &= -log \sum_k e^{\log(c_k) - \frac{1}{2}\sum_{t=1}^{T_{pred}} \left(x_t^{gt} - x_{k,t}\right)^2 + \left(y_t^{gt} - y_{k,t}\right)^2} \end{aligned} \tag{16}$$

where $N(.;\mu,\ \sigma)$ is the probability density function for the multivariate Gaussian distribution with mean μ and covariance matrix σ, X^{gt} is the given ground truth trajectory, $x_{k,t}$ and $y_{k,t}$ are the horizontal and vertical coordinates, respectively, of the kth possible trajectory at moment t.

3. Experiments

This section presents the experimental details and results of the proposed model.

3.1 Data set description

The Waymo Open Motion Dataset is used to evaluate the proposed method. It consists of 103,354 fragments, each containing a 20-s 10 Hz object trajectory and map data for the area covered by that fragment. The fragments are further divided into 1 s of historical data and 8 s of future data. The task is to use the one-second historical trajectory of the target agent to predict trajectories of surrounding traffic agents in the next eight seconds. Follow the requirements of the Waymo Open Motion Dataset, we use the previous second, including the first 11 frames of data, to predict the next 80 frames of data.

3.2 Evaluation methodology

- Metrics: We adopt four standard metrics in meter: minimum Average Displacement Error (mADE), minimum Final Displacement Error (mFDE), miss rate (MR) and overlap rate. The mADE denotes the minimum average L2 distance between the ground truth and the predicted results of all time steps. The mFDE denotes the minimum average L2 distance at the final time step. The MR is defined as the state when none of the individual predictions are within a given threshold of the ground truth trajectory, and the overlap rate is computed as the rate at which the predicted trajectories overlap with any other objects.
- Baselines: We compare the proposed method with the following models, including *Basic_LSTM, Loft, LSTM_CV, AS_LSTM, ANET* and *AE_LSTM*, where *Basic_LSTM* is the result obtained after using the proposed data processing method instead of the one on the leaderboard.

3.3 Implementation details

3.3.1 Data processing

In the processing of data, we take each target agent as the center and standardize the lane and other agents in the scene. In addition, lane line points are sampled at 2-m intervals.

3.3.2 Model parameters

In the social aggregator, we use a single-layer MLP with a hidden dimension of 256 for the encoding of agents. The hidden dimension in the graph convolutional network used for interaction fusion and the subgraph used for lane encoding is 256. Because the number and length of roads in each map are different, for the convenience of training, we take the maximum number of roads and the longest length of all scenes in each batch size as the number and length of roads in the whole batch and fill in the extra part with 0 to unify the input data dimension. The hidden dimension of LSTM in decoder is 512, and two two-layer MLPs are used to predict K possible trajectories and the corresponding confidence levels, respectively. In our experiments, K is set to 6.

3.3.3 Training details

During the training process, the initial learning rate is set to 0.001 and reduced at every 10,00 steps with max training epochs of 100. And the training dataset is 35 Waymo Open Motion Dataset, with a total data size of 20.5 G.

3.4 Results on benchmarks

The experimental results are shown in Table 1. Compared with trajectory prediction methods based on LSTM, the proposed methods have advantages in most of the metrics, and there is a significant improvement compared with *Basic_LSTM*. The social relation and physical lane aggregator can provide a good complement to the LSTM-based trajectory prediction and achieve better results.

Figure 4 shows the prediction results for several cases, including pedestrians and vehicles, straight ahead or turning at intersections. The figure shows the target agents' six possible trajectories, and the polyline with red dots is the most likely trajectory. The red line segments in each subfigure represent past trajectories of agents. It can be seen that our proposal is effective for the prediction of trajectory; whether it is in a more interactive intersection or straight road, it can predict the trajectory in a certain time period. Figure 5 shows the convergence speed of the proposed method and *Basic_LSTM*. It is obvious that our method converges faster.

3.5 Ablation experiments

To verify the effectiveness of the social aggregator and hybrid interaction aggregator module, we conducted ablation

Table 1 Model performance on the Waymo Open Dataset

Methods	mADE	mFDE	MR	Overlap Rate
Basic_LSTM	5.5092	11.5927	0.8034	0.2937
Loft	6.1850	14.5074	0.8237	0.2697
LSTM_CV	4.6984	11.1285	0.7831	0.3145
AS_LSTM	4.5373	10.8299	0.7374	0.2714
ANET	4.0467	9.8998	0.7939	0.2824
AE_LSTM	3.9917	10.8874	0.7490	0.2761
Raster_MP	3.4021	8.7416	0.6974	0.2367
Social-Lane Aggregator (Ours)	3.0452	6.1735	0.5674	0.2615

Figure 4 Trajectories generated by social-lane aggregator

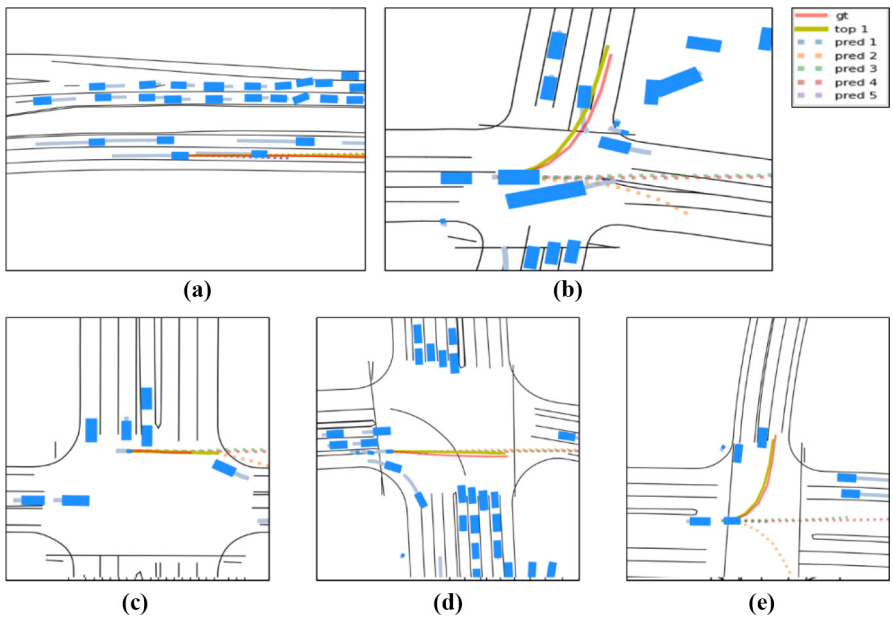
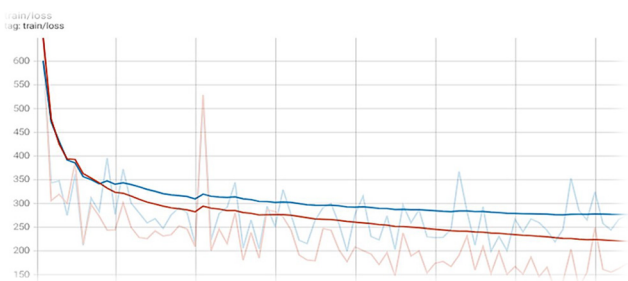


Figure 5 Model convergence speed comparison



experiments on the adjacency matrix and road feature extraction parts, and the results are shown in Table 2.

From Table 2, we can see the proposed modules contribute to improving the prediction performance. It can be seen that the mADE is improved from 4.6 to 4.2 by applying the attention-based adjacency matrix and from 4.6 to 3.3 by applying the hybrid interaction aggregator. With the contextual information being captured and incorporated by each module, the model has a more detailed understanding of the whole scene. For example, the hybrid interaction aggregator captures road information, incorporates useful road features, and thus, has a large lift (about 28.26%). When using the attention-based adjacency matrices instead of just distance relationships, more interaction features can be captured, resulting in an 8.31% improvement.

4. Conclusion

In this paper, the social relation and physical lane aggregator, which includes a social aggregator and hybrid interaction aggregator based on GCN and the self-attention mechanism, is proposed to explore and obtain social and physical interactions of traffic agents and the road environment. To validate the effectiveness, the proposed method is tested on the Waymo Open Motion Dataset and achieves better results on trajectory prediction. In verifying the computation of the adjacency matrix in GCN, the distance function and the attention mechanism were explored and compared, and the results proved that the adjacency matrix based on the attention mechanism better describes the relationship between agents; also, the ablation experiments were conducted for the two main modules that we designed. In this experiment, we assumed that

Table 2 Ablation on social-lane aggregator

Distence Adjacency	Attention Adjacency	Hybrid Interaction Aggregator	mADE	mFDE
✓			4.6710	8.8616
	✓		4.2967	8.1718
✓		✓	3.3117	6.3278
	✓	✓	3.0452	6.1735

agents and lane rules are equally important to drivers, but this may not be the case. Future work can continue to verify which is more influential, agents or lane lines.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L. and Savarese, S. (2016), "Social LSTM: human trajectory prediction in crowded spaces", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961-971.

Chen, W.C., Yu, X.Y. and Ou, L.L. (2022), "Pedestrian attribute recognition in video surveillance scenarios based on view-attribute attention localization", *Machine Intelligence Research*, Vol. 19 No. 2, pp. 153-168.

Deo, N. and Trivedi, M.M. (2018), "Convolutional social pooling for vehicle trajectory prediction", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1468-1476.

Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C. and Schmid, C. (2020), "Vectornet: encoding hd maps and agent dynamics from vectorized representation", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11525-11533.

Gu, J., Sun, C. and Zhao, H. (2021), "DenseTNT: end-to-end trajectory prediction from dense goal sets", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15303-15312.

Konev, S., Brodt, K. and Sanakoyeu, A. (2022), "MotionCNN: a strong baseline for motion prediction in autonomous driving", *in Workshop on Autonomous Driving*, CVPR.

Li, L., Zhou, B., Ren, W. and Lian, J. (2021), "Review of pedestrian trajectory prediction methods", *Chinese Journal of Intelligent Science and Technology*, Vol. 3 No. 4, pp. 399-411.

Lv, Y., Duan, Y., Kang, W., Li, Z. and Wang, F.-Y. (2014), "Traffic flow prediction with big data: a deep learning approach", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16 No. 2, pp. 865-873.

Shi, L., Wang, L., Long, C., Zhou, S., Zhou, M., Niu, Z. and Hua, G. (2021), "SGCN: sparse graph convolution network for pedestrian trajectory prediction", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8994-9003.

Song, X., Chen, K., Li, X., Sun, J., Hou, B., Cui, Y. and Wang, Z. (2020), "Pedestrian trajectory prediction based on deep convolutional LSTM network", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22 No. 6, pp. 3285-3302.

Wang, F.-Y. (2010), "Parallel control and management for intelligent transportation systems: concepts, architectures, and applications", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11 No. 3, pp. 630-638.

Wang, F.-Y., Zheng, N.N., Cao, D., Martinez, C.M., Li, L. and Liu, T. (2017), "Parallel driving in CPSS: a unified approach for transport automation and vehicle intelligence", *IEEE/CAA Journal of Automatica Sinica*, Vol. 4 No. 4, pp. 577-587.

Wei, Z., Li, Z., Wang, C., Chen, Y., Miao, Q., Lv, Y. and Wang, F.-Y. (2021), "Recurrent attention unit: a simple and effective method for traffic prediction", *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*.

Zhang, T., Song, W., Fu, M., Yang, Y. and Wang, M. (2021), "Vehicle motion prediction at intersections based on the turning intention and prior trajectories model", *IEEE/CAA Journal of Automatica Sinica*, Vol. 8 No. 10, pp. 1657-1666.

Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B. and Anguelov, D. (2020), "TNT: target-driveN trajectory prediction", arXiv preprint arXiv:2008.08294.

Further reading

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P. and Anguelov, D. (2020), "Scalability in perception for autonomous driving: Waymo open dataset", *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446-2454.

Waymo open dataset motion prediction challenge: Leaderboard (2021), https://waymo.com/open/challenges/2021/motion-prediction

Corresponding author

Yisheng Lv can be contacted at: yisheng.lv@ia.ac.cn