Relational Recurrent Neural Networks For Vehicle Trajectory Prediction

Kaouther Messaoud, Itheri Yahiaoui, Anne Verroust-Blondet, Fawzi Nashashibi

▶ To cite this version:

Kaouther Messaoud, Itheri Yahiaoui, Anne Verroust-Blondet, Fawzi Nashashibi. Relational Recurrent Neural Networks For Vehicle Trajectory Prediction. Intelligent Transportation Systems Conference (ITSC), Oct 2019, Auckland, New Zealand. 10.1109/ITSC.2019.8916887 . hal-02195180

HAL Id: hal-02195180

https://inria.hal.science/hal-02195180

Submitted on 5 Aug 2019

Relational Recurrent Neural Networks For Vehicle Trajectory Prediction

Kaouther Messaoud[1], Itheri Yahiaoui[2], Anne Verroust-Blondet[1] and Fawzi Nashashibi[1]

Abstract— **Scene understanding and future motion prediction of surrounding vehicles are crucial to achieve safe and reliable decision-making and motion planning for autonomous driving in a highway environment. This is a challenging task considering the correlation between the drivers behaviors. Knowing the performance of Long Short Term Memories (LSTMs) in sequence modeling and the power of attention mechanism to capture long range dependencies, we bring relational recurrent neural networks (RRNNs) to tackle the vehicle motion prediction problem. We propose an RRNNs based encoder-decoder architecture where the encoder analyzes the patterns underlying in the past trajectories and the decoder generates the future trajectory sequence. The originality of this network is that it combines the advantages of the LSTM blocks in representing the temporal evolution of trajectories and the attention mechanism to model the relative interactions between vehicles. This paper compares the proposed approach with the LSTM encoder decoder using the new large scaled naturalistic driving highD dataset. The proposed method outperforms LSTM encoder decoder in terms of RMSE values of the predicted trajectories. It outputs an estimate of future trajectories over 5s time horizon for longitudinal and lateral prediction RMSE of about 3.34m and 0.48m, respectively.**

I. INTRODUCTION

For a safe and efficient navigation, autonomous vehicles need to acquire the ability to analyze and understand different driving situations. They require information about the future intentions of surrounding vehicles in order to assess the driving situation and decide about their own future trajectories accordingly. Predicting the trajectory of a vehicle is a challenging task since it is highly correlated to other drivers' behaviors. Many studies tackle this task using traditional data-driven techniques [1], [2], [3] as well as deep learning models [4], [5], [6], [7], [8], [9], [10]. LSTMs have shown great success in modeling temporal data. Therefore, recent studies [9], [11], [12] use an LSTM based encoder decoder architecture to model the spatial interactions between neighboring vehicles. However, LSTMs lack the spatio-temporal structure to capture both, the temporal evolution and the spatial interactions between vehicles in the driving scene.

As a remedy, this paper proposes the use of a new architecture based on human like reasoning which selectively focuses attention on a subset of surrounding vehicles and efficiently retain pieces of information that probably influence his future trajectory. For instance, a driver intending to make a lane change focuses more on the vehicles in the target lane. Therefore, its future trajectory can be more influenced by distant vehicles in the target lane than the close ones in the other lanes.

The proposed architecture is based on Relational Recurrent Neural Networks (RRNNs) [13] encoder decoder. It combines the advantages of LSTMs in sequence modeling and the power of attention mechanism to capture the spatial inter-vehicles interactions. It is characterized by:

- **Per block information storing:** Input information are selectively stored into separate interacting blocks based on their content.
- **Relational reasoning:** Some vehicles are more likely to be related to or influenced by the other vehicles because of some features.
- **No distance constrained analysis:** Dependence between vehicles is not always tied to proximity in space.
- **Different focusing:** Different relations are encoded based on selective attention to a set of input information.

We use the new publicly available naturalistic vehicle trajectory highD dataset [14] to train and validate our model in the task of trajectory prediction. Therefore, we compare our model to LSTM based encoder decode model and we provide better results in terms of longitudinal and lateral prediction RMSE.

II. RELATED RESEARCH

In their surveys, Lefèvre et al. [15] and Zhan et al. [16] divide the vehicle behavior forecasting methods into two main categories based on whether they consider the interactions between the neighboring vehicles or not.

A. Independent prediction

Independent vehicle motion prediction approaches consider, in their model, only one single vehicle at a time. Early work predicted future trajectories based on physics evolution models like Switching Kalman Filters [17], Constant Turn Rate and Velocity model (CTRV) [18], Interacting Multiple Models [19] and Intelligent Driver Model (IDM) [20]. They mainly rely on the low level characteristics of motion. Therefore, they are constrained to short-term motion prediction. More recent methods decompose the motion of a vehicle into a set of patterns or maneuvers. They consider motion prediction as a multi class classification problem then use the predicted maneuvers to infer the future trajectory [2]. Yoon et al. [21] base their motion prediction on the vehicles target lane and propose three representative trajectories per lane depending on how fast the vehicle attain that lane. They use the Multi-Layer Perceptron MLP to estimate the probabilities of each lane and each of the possible trajectories.

[1] Inria Paris, 2 rue Simone Iff 75012 Paris FRANCE {kaouther.messaoud,anne.verroust,fawzi.nashashibi}@inria.fr

[2] CReSTIC, Université de Reims Champagne-Ardenne, Reims, FRANCE itheri.yahiaoui@univ-reims.fr

These models are constrained as they do not consider the influence of the neighboring vehicles on the predicted trajectory.

B. Interaction aware prediction

1) Inverse Reinforcement Learning (IRL): Drivers decision-making process can be considered as a Markov Decision Process (MDPs): Each vehicle, when it moves, it minimizes a cost function. Sierra Gonzlez et al. [22] deploy an IRL algorithm to infer the cost function parameters. Then, they merge it with a heuristic policy model to present the risk-aversive behavior of drivers. They predict the future motion by sequentially applying the actions estimated by this policy. In [23], they combine the driver model with Dynamic Bayesian Networks (DBN) to represent interactions between vehicles.

2) Recurrent Neural Networks (RNNs): Recent advancements in sequence modeling is a result of the use of recurrent neural networks (RNNs). They have shown promising results in diverse domains such as natural language processing (NLP) and speech recognition. Long Short Term Memories (LSTMs) are particular implementations of RNNs. They propose to model long-term dependencies between input features. Therefore, they operate by storing, and retrieving information to learn to relate inputs. Therefore, LSTM based approaches have been solid candidates to model maneuver and trajectory prediction.

LSTMs have been recently deployed for driver intention prediction. Different LSTM-based approaches have been used; A simple LSTM with one or more layers was utilized in [5], [6], [7], [10]. Xin et al. [8] use a dual LSTM. The first one for high-level driver intention recognition succeeded by a second generating the corresponding predicted trajectory. Others [9], [11] deploy an LSTM encoder decoder architecture. Different input features are tested. While Lenz et al. [6] inputs to the LSTM only the current state of the target and a set of its surrounding vehicles in order to match the Markov Property, other studies [5], [7], [9] consider the sequence of past features to provide the model with the temporal evolution patterns and improve the trajectory prediction. They attribute to the LSTM the mission of retaining the relevant events and considering them to generate the predicted trajectory.

Coming to modeling the interactions between surrounding vehicles, most of existing models [5], [6], [7], [9] implicitly infer the dependencies between vehicles. They let the LSTM implicitly learn the influence of surrounding vehicles on the target vehicle's motion by introducing a sequence of surrounding vehicles features as inputs to the LSTMs. LSTMs compress all the received track sequence into a common hidden vector. This can limit its performance in modeling the inter-vehicles dependencies.

Attention mechanisms and mainly self-attention [24], have been used in a lot of novel neural network architectures [24], [25], [26] due to their good performance at capturing long range dependencies. Additionally, they reduce the number of local operations by directly relating distant elements.

In this work, we predict the future trajectory of a target vehicle by combining the advantages of LSTMs in sequence modeling and the power of attention mechanism to capture the spatial inter-vehicles dependencies. To that end, we bring relational networks based methods to the problem of interaction aware vehicle motion prediction. RRNNs extend the LSTM architecture by introducing interactive memory blocks using Multi-Head Dot Product Attention inside of the LSTM block.

Our motion prediction results are compared with LSTM based encoder decoder model.

III. PROBLEM DEFINITION

We aim to predict the future positions of a target vehicle T knowing its track history and the track history of its surrounding vehicles at current time t_{obs}.

A. Inputs and Outputs

We assume that we have as input the track history of the target and n surrounding vehicles. The input trajectory of a vehicle i is defined as $\mathbf{X}_i = [\mathbf{x}_i^1, \ldots, \mathbf{x}_i^{t_{obs}}]$ where $\mathbf{x}_i^t = (x_i^t, y_i^t)$. We note (x_T^t, y_T^t) the coordinates of the target vehicle T.

The coordinates are expressed in a stationary frame of reference where the origin is the position of the target vehicle at time t_{obs}. The $y-axis$ and $x-axis$ point respectively to one direction of motion of the freeway and to the direction perpendicular to it.

We define a $3D$ spatial grid H^t composed of the coordinates of the target and its surrounding vehicles at time t based on their positions at time t_{obs}.

$$H^t(m, n, :) = \delta_{mn}(x_i^{t_{obs}}, y_i^{t_{obs}})(x_i^t, y_i^t) \qquad \forall i \in \mathcal{A}_T \qquad (1)$$

$\delta_{mn}(x, y)$ is an indicator function equal to 1 if and only if (x, y) is in the cell (m, n), $\mathcal{A}_T$ is the set of neighboring vehicles in addition to the target one.

The columns correspond to the three lanes. We consider a grid size of $(13, 3)$ centered on the target vehicle position and covering a longitudinal distance of 58.5 meters (Grid cell size $= 4.5m$).

Unlike most of the state of the art works that consider the vehicles immediately around the target vehicle, we adopt a grid over the neighboring area. This representation of the scene has the following advantages:

- It models the spatial distances between the vehicles in the scene and represents the drivable areas.
- It enables us to consider different scenarios with different numbers of traffic participants.
- It preserves the lane structure of the highway.

The output of the model is the sequence of the target vehicle's predicted future positions.

$$\mathbf{Y}_{pred} = [\mathbf{y}_{pred}^{t_{obs}+1}, \ldots, \mathbf{y}_{pred}^{t_{obs}+t_f}]$$

Where $\mathbf{y}_{pred}^t = (x_{pred}^t, y_{pred}^t)$ is the target vehicle's predicted coordinates.

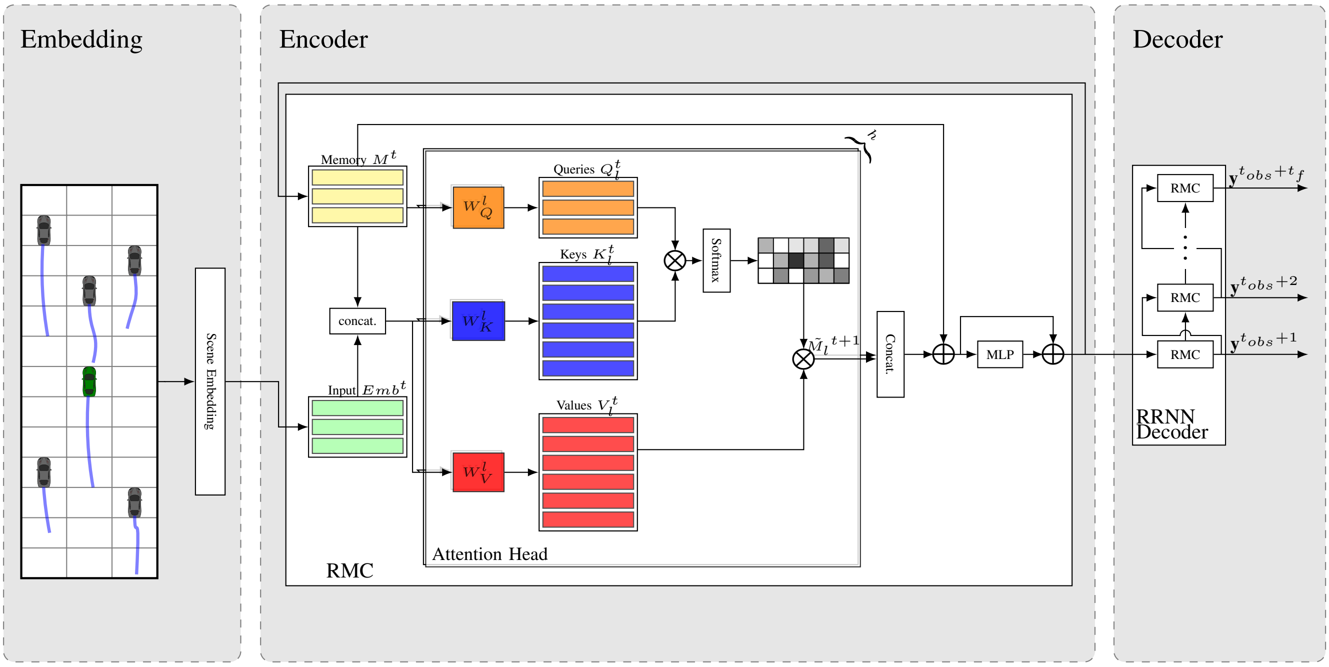

Fig. 1. Proposed Model (Per lane scene embedding L-RRNN Example)

B. Loss Function

We train the model first by minimizing the root mean squared error between the real trajectory and the predicted one:

$$L_{RMSE} = \sqrt{\frac{1}{t_f} \sum_{t=t_{obs+1}}^{t_{obs}+t_f} (x_T^t - x_{pred}^t)^2 + (y_T^t - y_{pred}^t)^2} \tag{2}$$

IV. MODEL ARCHITECTURE

Fig. 1 shows our proposed model. It consists of a scene embedding (cf. IV-C) and RRNNs based encoder and decoder (cf. IV-A). It illustrates the per lane scene embedding **L-RRNN** described in IV-C.2. After the scene grid embedding, the encoder learns the vehicle motion and captures the dependencies in the input data using the Relational Memory Core (RMC) block. For each iteration, the RMC is fed with the previous memory matrix M^t and the current scene embedding Emb^t. It applies the Multi-Head Dot Product Attention (MHDPA) (cf. IV-B) to provoke the interaction between memory and input slots. MHDPA operates by projecting each memory and input slot using row-wise shared weights W_Q^l, W_K^l and W_V^l to generate the queries Q^t, keys K^t and values V^t respectively. The MHDPA module is followed by row-wise multilayer perceptron MLP, then, the resultant memory is gated to form the next memory state and an output vector which are fed to the decoder at t_{obs}. The decoder, composed of RRNNs, outputs the predicted future trajectory of the target vehicle.

A. Relational Recurrent Encoder-Decoder

We deploy an encoder decoder architecture in the task of trajectory prediction:

- **RRNN encoder:** receives the input sequence embedding, extracts the properties of the target vehicle past trajectory and interaction information, compresses them in an encoding vector and feeds this vector with the memory block to the decoder.
- **RRNN decoder:** learns to generate the predicted trajectory based on the received information: At t_{obs} time step, the decoder has as input the encoding vector and the memory block. It makes prediction for the next time steps and generates the next memory block. Then, we proceed by forming and passing the memory blocks and reinjecting the decoder's predictions into the decoder's input of the next time step to sequentially generate the predicted target vehicle positions.

The encoder and decoder are composed of RRNNs. RRNNs are memory based recurrent neural networks able to perform relational reasoning between input entities over time. They are based on iterative information selective storing into blocks and computing interactions between them. In fact, each RRNN block contains a number of memory slots where the pertinent information are stored.

RRNN operate by slicing the memory and the input into slots and heads and provoking interactions between them. Indeed, each memory slot is updated each time step based on:

- **Memory-Memory attention:** each memory slot attends over the other memory slots. This captures the interactions and dependencies in the stored information.
- **Memory-Input attention:** each memory slot attends over the input embedding slots. Attention enables to decide which information from the input would be stored in adequate memory slots based on its relation to what is already contained in the memory. This infer

inter-vehicles interactions as well.

B. Multi-Head Dot Product Attention (MHDPA):

In each RRNN block, we use linear projections of the previous memory M^t and the input embedding Emb^t at each time step t to generate the queries $Q_l^t = M^t W_Q^l$, keys $K_l^t = [M^t; Emb^t] W_K^l$ and values $V_l^t = [M^t; Emb^t] W_V^l$. $[M^t; Emb^t]$ denotes the row-wise concatenation of M^t and Emb^t.

In order to enable the memory slots to share different information and represent different interactions, we use multiple attention heads. Therefore, we generate h sets of queries, keys, and values for $l = 1..h$ using different projection matrices.

The memory is updated using multi-head dot product attention over the other memory slots and the current input embedding:

$$\tilde{M_l}^{t+1} = A(Q_l^t, K_l^t, V_l^t) = \underbrace{softmax(\frac{Q_l^t transpose(K_l^t)}{\sqrt{d^k}})}_{attention\ weights} V_l^t \tag{3}$$

$\tilde{M}_l^{t+1}$ is an update of the memory where each slot is a weighted sum of the projections of the previous memory slots and the projections of the current embedding input.

d^k is a a scaling factor that corresponds to the dimensionality of the key vectors.

We apply the attention operation described above for each head. The resulting memory $\tilde{M}^{t+1}$ is column-wise concatenation of the memories $\tilde{M}_l^{t+1}$ for $l = 1..h$.

We employ a residual connection [27] around the MHDPA followed an MLP then a second residual connection. These operations are encapsulated into an LSTM cell as described in [13]. Therefore, the resultant memory block is gated and used as next memory state M^{t+1}.

C. Inputs Embedding

In this work, we use two different ways of embedding the input data, and then we compare the results of the different methods.

1) Scene embedding (Sc-RRNN): We consider the whole scene as an input vector. We embed the scene using a fully connected layer to generate an embedding vector. The vectors embedding the scene for time steps $t = 1, \ldots, t_{obs}$ are sequentially fed to the RRNN encoder:

$$Emb^t = \Psi(H^t; W_{emb})$$

The RRNN implicitly infers the interactions and the dependencies between the input vehicles.

2) Per lane embedding (L-RRNN): We divide the scene based on lanes to generate an input matrix. We embed each lane using a fully connected layer Ψ to generate an embedding matrix of size $(3, size\ of\ embedding)$. The matrix embedding the scene for time steps $t = 1, \ldots, t_{obs}$ are sequentially fed to the RRNN encoder:

$$Emb^t(n,:) = \Psi(H^t(:,n,:); W_{emb}),\ n = 1,2,3$$

This model conserves the lane-wise structure of the road. It captures the spatio-temporal interactions between vehicles in the same and adjacent lanes. It performs a lane-based attention to focus on the lane-changing behavior.

In this model, we consider three memory slots to store lane-level information.

D. Training and Implementation Details

The input grid is embedded into an embedding vector or matrix of sizes 64 and (3,64) depending on the input embedding type. Then, we use the Leaky ReLU activation function with $\alpha = 0.1$.

We deploy RRNNs encoder decoder with two memory slots for Sc-RRNN and three for L-RRNN. Each memory slot is 64 in size. We employ $h = 2$ parallel attention heads over projected vectors of size 32. We use a batch size of 128. We adopt the Adam optimizer [28]. The model is implemented using PyTorch [29].

V. EXPERIMENTAL EVALUATION

A. Dataset

We are the first to use the new publicly available naturalistic vehicle trajectory highD dataset [14] in the task of trajectory prediction. Previous studies used even personal dataset or the Next Generation Simulation (NGSIM) [30], [31] dataset. However, Coifman et al. [32] prove annotations inaccuracies in the NGSIM dataset. This may result in physically unrealistic vehicle behaviors. Besides, highD is bigger than NGSIM. It contains about 12 times as many vehicles as NGSIM. Therefore, we choose the highD dataset to train and evaluate our network.

HighD [14] is a new dataset captured in 2017 and 2018. It is recorded by camera-equipped drones from an aerial perspective of six different German highways at 25 Hz. It is composed of 60 recordings of about 17 minutes each, covering a segment of about 420m of two driving directions roads (Figure 2). It consists of vehicle position measurements from six different highways with 110 000 vehicles and a total driven distance of 45 000 km. This dataset is important since it has 5 600 recorded complete lane changes and presents recent driver behaviors.

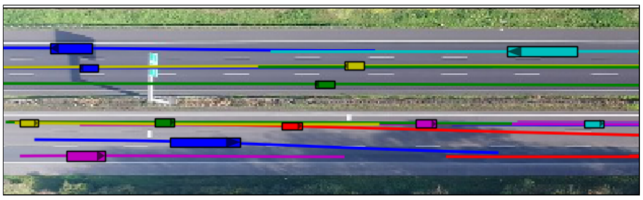

Fig. 2. Highway drone dataset highD [14]

We split each of the 60 recordings of the highD dataset into train (75%) and test (25%) sets. Therefore, we include different driving behaviors at different times of the day and different locations in both train at test sets. This enhance the network ability to learn generalized behavior over different drivers and different driving conditions. Then, we split the

TABLE I

Root mean squared prediction error (RMSE) in meters over a 5 second prediction horizon for the models.

Error	Prediction Horizon (s)	V-LSTM	Sc-LSTM	Sc-RRNN	L-RRNN
Total	1	0.31	0.32	0.29	**0.22**
	2	0.81	0.82	0.69	**0.65**
	3	1.51	1.60	1.33	**1.31**
	4	2.48	2.63	2.22	**2.22**
	5	3.71	3.87	**3.33**	3.38
Lateral	1	0.10	0.10	0.08	**0.05**
	2	0.32	0.20	0.18	**0.14**
	3	0.46	0.33	0.30	**0.26**
	4	0.57	0.45	0.43	**0.37**
	5	0.65	0.56	0.53	**0.48**
Longitudinal	1	0.27	0.31	0.27	**0.22**
	2	0.74	0.79	0.66	**0.63**
	3	1.44	1.57	1.30	**1.29**
	4	2.42	2.59	**2.16**	2.19
	5	3.65	3.83	**3.27**	3.34

trajectories into segments of 8s composed of a track history of 3s and a prediction horizon of 5s. We downsample each segment to get only 5 fps to reduce the complexity of the model.

B. Evaluation Metric

We use the predicted trajectories for the Root of the Mean Squared Error (RMSE) calculation. The RMSE averages the distance between predicted positions and the ground truth. We consider also the longitudinal and lateral errors to be able to infer further information about the error on lane change prediction.

C. Models Compared

We compare our proposed models with LSTM based encoder decoder architecture. For a fair comparison, we consider an LSTM having the same total memory size of the relational memory (Sc-RRNN) we have used.

- **Vanilla LSTM** (*V-LSTM*): an encoder decoder LSTM based model. It uses the track history of the target vehicle in the encoder LSTM and generates the output trajectory with the LSTM decoder. This represents an independent trajectory prediction model.
- **Scene LSTM Encoder Decoder** (*Sc-LSTM*): an encoder decoder based model where the encoder encodes the trajectories of the target and surrounding vehicles. The encoding vector is fed to the decoder which generates trajectory predictions.
- **Relational Recurrent Neural Network with scene embedding** (*Sc-RRNN*) model described in this paper.
- **Relational Recurrent Neural Network with per lane embedding** (*L-RRNN*) model described in this paper.

D. Results

Table I shows the RMSE values for the models being compared. First, we observe that Sc-LSTM and V-LSTM comparable total RMSE error. While Sc-LSTM produces better lateral error, it has larger longitudinal error than V-LSTM. This can imply that the LSTM has limited capability in capturing the effects of surrounding vehicles on predicting the future motion of the target vehicle. This also proves the effectiveness of considering neighboring vehicles in the prediction of the lateral motion of the target vehicle.

Both proposed methods, Sc-RRNN and L-RRNN, lead to further improvement in prediction error, suggesting the importance of the use of the multiple memory slots and the attention across these memories in the task of motion prediction. We also note that the improvement produced by the use of RRNNs seem to be more remarkable for longer prediction horizons. This implies that LSTM has limited capacity to perform long term relational reasoning.

Additionally, the per-lane embedding of the scene has produced lower lateral error. This can infer that explicit lane-wise division of the scene and the memory-input slots interactions via MHDPA give additional information about inter-lane dependencies. However, we assume that considerable further analysis of the architecture is needed using other metrics able to evaluate lane changes detection for example to conclude about the best way to embed the scene. Besides, the memory update should be studied over time for additional proof of the model performance.

VI. CONCLUSIONS

In this work, we presented a novel way to tackle the task of long-term (5s) trajectory prediction on highway using relational recurrent neural networks RRNNs. This approach combines the advantages of multi-head dot product attention mechanism and LSTMs to capture the spatio-temporal dependencies between the input tracks. The proposed model provided competitive results with the state-of-the-art on the naturalistic driving large scale highD dataset based on the RMSE metric for both longitudinal and lateral position prediction.

The deployed architecture represents a promising way for motion prediction of surrounding vehicles for autonomous vehicle. We believe that it can be extended and utilized to further improve vehicle motion prediction in various driving scenarios such as intersections and roundabouts. Moreover, and part of our future work, we plan to extend and validate the proposed approach to consider heterogeneous and mixed traffic scenarios with different road agents such as buses, trucks, cars, scooters, bicycles, or pedestrians.

ACKNOWLEDGMENT

The work presented in this paper has been financially supported by PIA French project CAMPUS (Connected Automated Mobility Platform for Urban Sustainability).

REFERENCES

[1] J. Schlechtriemen, A. Wedel, J. Hillenbrand, G. Breuel, and K. Kuhnert, "A lane change detection approach using feature ranking with maximized predictive power," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 108–114.

[2] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 4363–4369.

[3] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A unified framework for maneuver classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129–140, June 2018.

[4] A. Khosroshahi, E. Ohn-Bar, and M. M. Trivedi, "Surround vehicles trajectory analysis with recurrent neural networks," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 2267–2272.

[5] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 1665–1670.

[6] D. Lenz, F. Diehl, M. T. Le, and A. Knoll, "Deep neural networks for markovian interactive scene prediction in highway scenarios," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 685–692.

[7] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 353–359.

[8] L. Xin, P. Wang, C. Chan, J. Chen, S. E. Li, and B. Cheng, "Intention-aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 1441–1446.

[9] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 1179–1184.

[10] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 1484–1489.

[11] 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," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 1672–1678.

[12] H. Misawa, K. Takenaka, T. Sugihara, H. Liu, T. Taniguchi, and T. Bando, "Prediction of driving behavior based on sequence to sequence model with parametric bias," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 1–6.

[13] A. Santoro, R. Faulkner, D. Raposo, J. Rae, M. Chrzanowski, T. Weber, D. Wierstra, O. Vinyals, R. Pascanu, and T. Lillicrap, "Relational recurrent neural networks," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 7299–7310.

[14] 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," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 2118–2125.

[15] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, pp. 1–14, 2014.

[16] W. Zhan, A. L. de Fortelle, Y. Chen, C. Chan, and M. Tomizuka, "Probabilistic prediction from planning perspective: Problem formulation, representation simplification and evaluation metric," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 1150–1156.

[17] H. Veeraraghavan, N. Papanikolopoulos, and P. Schrater, "Deterministic sampling-based switching kalman filtering for vehicle tracking," in *2006 IEEE Intelligent Transportation Systems Conference*, Sep. 2006, pp. 1340–1345.

[18] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 549–562, Sep. 2007.

[19] R. Toledo-Moreo and M. A. Zamora-Izquierdo, "Imm-based lane-change prediction in highways with low-cost gps/ins," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 180–185, March 2009.

[20] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *2012 IEEE Intelligent Vehicles Symposium*, June 2012, pp. 1162–1167.

[21] S. Yoon and D. Kum, "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 1307–1312.

[22] D. Sierra González, J. S. Dibangoye, and C. Laugier, "High-speed highway scene prediction based on driver models learned from demonstrations," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 149–155.

[23] D. Sierra González, V. Romero-Cano, J. S. Dibangoye, and C. Laugier, "Interaction-aware driver maneuver inference in highways using realistic driver models," in *2017 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2017, pp. 1–8.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems (NIPS)*, 2017.

[25] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[26] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *CoRR*, vol. abs/1805.08318, 2018. [Online]. Available: http://arxiv.org/abs/1805.08318

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. D. L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS 2017 Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques*, Dec. 2017.

[30] J. Colyar and J. Halkias, "Us highway 101 dataset." in *Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT07-030*, 2007.

[31] J. Colyar and J. Halkias, "Intersate 80 freeway dataset." in *Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-06-137*, 2006.

[32] B. Coifman and L. Li, "A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset," *Transportation Research Part B: Methodological*, vol. 105, pp. 362–377, 11 2017.