

Received December 18, 2019, accepted January 16, 2020, date of publication January 22, 2020, date of current version February 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968618

# A Dual Learning Model for Vehicle Trajectory Prediction

MAHROKH KHAKZAR<sup>ID</sup>, ANDRY RAKOTONIRAINY<sup>ID</sup>, ANDY BOND<sup>ID</sup>,  
AND SEPEHR G. DEHKORDI<sup>ID</sup>

Centre for Accident Research Road Safety-Queensland, Queensland University of Technology, Brisbane, QLD 4059, Australia

Corresponding author: Mahrokh Khakzar (mahrokh.khakzar@qut.edu.au)

This work was supported by the Queensland University of Technology (QUT), Brisbane, QLD, Australia.

**ABSTRACT** Automated vehicles and advanced driver-assistance systems require an accurate prediction of future traffic scene states. The tendency in recent years has been to use deep learning approaches for accurate trajectory prediction but these approaches suffer from computational complexity, dependency on a specific environment/dataset, and lack of insight into vehicle interactions. In this paper, we aim to address these limitations by proposing a Dual Learning Model (DLM) using lane occupancy and risk maps for vehicle trajectory prediction. To understand the spatial interactions of road users, make the model independent of the environment, and consider inter-vehicle distances, we embed an Occupancy Map (OM) into the trajectory prediction model. We also utilise a traffic scene Risk Map (RM) to explicitly consider a comprehensive definition of risk based on Time-to-Collision in the traffic scene. These two features employed in the encoder-decoder architecture improve system accuracy with less complexity and provide insight into the interaction between all road users. The experiment has been conducted on two different naturalistic highway driving datasets (i.e., NGSIM and HighD) demonstrating algorithm independence from a single environment. Comparison results indicate that the DLM achieves a more accurate trajectory prediction with a less complex structure compared with existing approaches in terms of RMS prediction error, which indicates the effectiveness of DLM in such a context.

**INDEX TERMS** Vehicle trajectory, trajectory prediction, recurrent neural network, deep feature learning, and long-short-term memory.

## I. INTRODUCTION

Each year, 1.35 million people die, and as many as 50 million are injured and experience long-term disability in road crashes. Road crashes are the 8th leading cause of death and the biggest killer of those aged 5–29 [1]. To reduce road crash fatalities and have a safer and more efficient transportation system, automated vehicles and driver-assistance systems have become a promising solution. Recent research has revealed the widespread significance of these technologies [2]–[5]. Despite substantial attention of researchers and industry, accurate prediction of vehicle trajectory in a dynamic traffic scene remains a priority, challenging problem due to its complex nature. In a reliable trajectory prediction system, accuracy is not the only indicator of system effectiveness. The horizon of prediction, ability to be generalised to

new traffic environments and consideration of future vehicle trajectory as influenced by the surrounding vehicle trajectories (inter-vehicle interaction) are also significant.

In this study, we propose a Dual Learning Model (DLM) for vehicle trajectory prediction that identifies salient information in an Occupancy Map (OM) and a Risk Map (RM) to learn about inter-vehicle interaction and associated risk within the traffic scene in an unsupervised manner. It is a data driven approach with two input channels, using a Long-Short-Term Memory (LSTM) encoder-decoder structure. This methodology is motivated by the success of LSTM networks in modelling non-linear temporal dependencies in sequence-to-sequence prediction. We have evaluated the proposed system with two distinct naturalistic highway driving scenarios which vary in their traffic flow and driving style to illustrate the ability of the proposed system to be extended for different environments. The contribution of this study can be summarised as follows:

The associate editor coordinating the review of this manuscript and approving it for publication was Kan Zheng<sup>ID</sup>.

- i. We inject the Risk Map into the trajectory prediction pipeline, where the proposed structure learns to automatically extract, store and retrieve salient information without any supervision.
- ii. We test the proposed model on two distinct naturalistic highway driving datasets, containing more than 108570 unique trajectories from different datasets, to prove the independence of the proposed system from a specific environment.
- iii. We extensively evaluate the performance of the proposed method with state-of-the-art methods. We show that our proposed method outperforms the state-of-the-art methods.

The rest of this paper is organised as follows: First, we provide a brief background to the problem including a literature review followed by a discussion of the problem formulation. Next, we introduce the structure of our proposed models in Section III. Then, we present the simulation result and discussion. Finally, a summary of our numerical results is followed by the concluding remarks.

## II. BACKGROUND AND LITERATURE REVIEW

Vehicle trajectory prediction is required in various areas of transportation such as automated vehicles and driving assistance systems. An extensive survey of vehicle trajectory prediction models classified trajectory prediction approaches into three categories; Physics-based models, Manoeuvre-based models, and Interaction-aware models [6].

Physics-based approaches have a short time prediction (i.e., less than a second). These approaches assume a constant speed and orientation for the vehicles and possess the lowest degree of abstraction in trajectory prediction. These applications are the most common and simplest techniques using dynamic and kinematic models for trajectory prediction [6]. One of the weaknesses of these systems is that they cannot predict change in vehicle motion caused by manoeuvres or change because of external factors, for instance change to front vehicle speed.

The second category is Manoeuvre-based approaches. These approaches assume that a vehicle performs a series of independent manoeuvres from the other vehicles on the road network. The rationale for this approach is twofold; first the vehicle manoeuvre is recognised and then the result of manoeuvre recognition modules is used to make better predictions of future trajectories [3], [7]. The manoeuvre recognition step is typically provided by classifiers such as Heuristic-based classifiers [8], Bayesian networks [7], hidden Markov models [3], [9], random forest classifiers [10] or recurrent neural networks [11], [12]. The output of the previous manoeuvre classification step is an input to the trajectory prediction step and the output of this module is the future location of the vehicle [3], [7]–[9], [13]. Although Manoeuvre-based approaches can decompose vehicle movements into several succinct manoeuvres that are relatively easy to recognise [6], in a complex scenario, it is hard to reasonably categorise vehicle manoeuvres into

several categories. Furthermore, trajectories must be manually labelled for training purposes, and more so, labelling tasks is time-consuming and labelling errors may increase the error of the trained model.

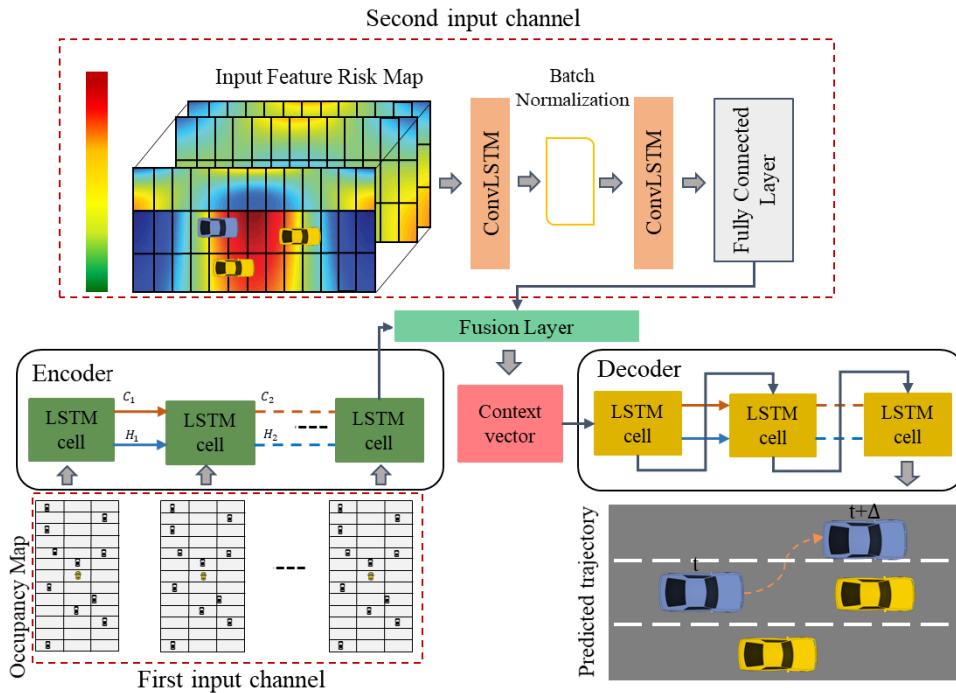
The third category is Interaction-aware models. These approaches consider the efficacy of inter-vehicle interaction on vehicle trajectory prediction. They are the most comprehensive approach providing more reliable long-term prediction. Coupled HMMs (CHMMs) can model pairwise dependency of motion in complex traffic situations [14]. However, when the number of possible pairwise dependencies increases quadratically with the number of entities, this complexity is not manageable. To solve this problem, Oliver and Pentland [15] proposed a solution to simplify the model by considering the individual impact of each vehicle surrounding the ego-vehicle,<sup>1</sup> thereupon CHMMs would be asymmetric and computational complexity reduced.

Interaction-aware models based on the method of considering inter-vehicle interaction can be broadly categorised into two domains: *handcrafted cost functions* and *end-to-end models* [4].

In the first, handcrafted cost functions are used based on the relative configuration of vehicles and predict future manoeuvres by considering these cost functions [3], [16]. Bahram, et al. [16] proposed a two-level interaction-aware manoeuvre prediction model. In the first level, the future manoeuvres of the surrounding vehicle are predicted by considering the road geometry, traffic rules and interaction between the vehicles in a traffic scene. Then, in the second level, a Manoeuvre-based classifier is used to learn different manoeuvre patterns, which in an unusual driving case, makes the prediction more robust against possibly inaccurate model assumptions. Although cost function-based methods can be generalised to new traffic conditions and do not rely on training data, they are bound to the design of the hand-crafted cost functions.

The second group is end-to-end models which learn from real traffic data. These models skip the manoeuvre recognition module and perform a direct trajectory prediction which can avoid errors in manoeuvre recognition [2]. Owing to the massive variation in traffic configurations, these approaches require a large dataset for generalisation [4]. With advances in artificial intelligence, various deep neural network (DNN) techniques have now been carried out on trajectory prediction. Among different methods of DNN, Recurrent Neural Networks (RNNs) and their variants such as the long-short-term memory (LSTM) model have proven to work effectively with time series data sequence generation. One of the most popular methods is using an encoder-decoder structure to encode and decode trajectory information. Park, et al. [17] proposed an encoder-decoder LSTM model for predicting vehicle trajectory by using an occupancy grid map [18] and a beam search algorithm which reduces the error propagation

<sup>1</sup>Ego-vehicle corresponds to the vehicle where the main observations take place.



**FIGURE 1.** Proposed Dual Learning Model (DLM) framework.

issue appearing in the iterative decoding step and keeps the best trajectories for each decoding iteration.

The maximum prediction horizon of this model is two seconds, however literature shows that this timeframe might not be sufficient for applications such as collision warning systems where the optimal timing of prediction is at least 4 seconds [19].

Deo and Trivedi [4] adopt a convolutional social pooling LSTM based model. This model contains an LSTM encoder which takes the most recent track history of the ego-vehicle and all the surrounding vehicles to learn vehicle motion dynamics. A convolutional social pooling layer extracts a social tensor which captures interdependencies of vehicles. Finally, a manoeuvre-based decoder predicts a distribution of future vehicle trajectories. This approach first predicts the manoeuvre of a vehicle and then predicts the trajectory dependent on that manoeuvre. Despite the fact that this method is not limited by the design of a hand-crafted cost function, data existence, and prediction horizon, the approach does not consider the impact of distance between road users (i.e., closer vehicles have a higher impact on an ego-vehicle trajectory). To deal with this important issue, Dai, et al. [2] proposed a Spatio-Temporal LSTM based model. The advantage of this work compared with the previous work is that this model considers the safe distance between surrounding road users to measure the degree of influence between them. However, to implicitly measure this important influence, a more comprehensive definition of safety must be considered.

Though there have been efforts to develop a variety of trajectory prediction approaches in recent years, they have not

taken into account either the associated risk within the traffic scene or the degree of influence of surrounding vehicles. Furthermore, most of the existing approaches are focused on a specific dataset making them impractical for more general application. To investigate this aspect of generality, we evaluated the system across NGSIM and HighD datasets. These datasets were collected in two different countries containing both Change Lane and lane-keeping (Go Straight) trajectories with different speed limits, driving culture and road geometry. The proposed model in this study adopts a different architecture called the Dual Learning Model (DLM) which takes information from two different inputs to predict vehicle trajectory without being specific to one dataset.

### III. METHODOLOGY AND MODEL DESIGN

In this section, we describe the problem formulation and our proposed approach. The problem we are addressing can be defined as predicting the trajectory of an ego-vehicle for a certain time horizon while its trajectory is influenced by other road user trajectories. Figure 1 illustrates the proposed framework in this study. The approach illustrated in Figure 1 uses an LSTM encoder-decoder structure for time series prediction allowing an insight into past trajectory values to evaluate how the system will evolve in the future and thus, predict future values. It means that the system remembers important past events and applies this learning to predict future values. We take the encoder-decoder architecture as the base model which is comprised of two sub-models; an encoder to remember important past events, and a decoder to convert the important events into a prediction for the future.

We employ LSTM networks due to their remarkable success in sequence-to-sequence prediction. We use a multimodal information fusion technique to inject relevant features into the encoder-decoder structure by applying a convolutional LSTM mechanism on top of this base model [20]. The system extracts the risk features of the traffic scene and models the possible risk interaction among vehicles in the traffic scene on the top of the context vector. Risk is represented by Time-to-Collision between vehicles in the traffic scene. Later, we will show that the proposed model provides a more accurate trajectory prediction result compared with the state-of-the-art.

In the following section, we describe the Encoder LSTM, Convolutional LSTM, Decoder LSTM, and input channels in detail.

#### A. ENCODER LSTM

The encoder receives multiple input sequences ( $X$ ) in the form of trajectory sequences from the ego-vehicle and encodes them to generate the encoded sequences  $h$ , which is given by

$$X = [\psi_{i-n}, \dots, \psi_{i-1}, \psi_i] \quad (1)$$

$$\psi_i = [x_i, y_i, v_{x,i}, v_{y,i}, a_{x,i}, a_{y,i}, W_{v,i}] \quad (2)$$

$$h_n = [h_1, \dots, h_i] \quad (3)$$

where  $n$  is the length of track history,  $\psi_i$  is the feature vector at each time step  $i$ . The feature sequences,  $\psi_i$ , is obtained from ego-vehicle kinematic information as well as surrounding vehicle information. In Equation 2,  $x_i$  and  $y_i$  are lateral and longitudinal coordinates of the ego-vehicle in the  $i^{th}$  time step where  $x_0 = y_0 = 0$ . The lateral velocity, longitudinal velocity, lateral acceleration, and longitudinal acceleration are shown by  $v_x$ ,  $v_y$ ,  $a_x$ , and  $a_y$ , respectively.  $W_{v,i}$ , in Equation (2), represents weights which reflect the degree of surrounding vehicle influence on the ego-vehicle trajectory. We will justify the use and impact of  $W_{v,i}$  on improving the accuracy of trajectory prediction later in this paper. The encoded sequence,  $h_i$ , is given by an LSTM encoding function, where  $X_i$  the is input sequence,  $h_{i-1}$  is a hidden state at time  $i - 1$  and  $c_i$  is the context vector at time  $i$ .

$$h_i = \text{LSTM}(X_i, h_{i-1}) \quad (4)$$

$$c_i = \tanh(h_i) \quad (5)$$

#### B. CONVOLUTIONAL LSTM (ConvLSTM) NETWORK

The aim of this section is to use the salient Risk Map information extracted by the ConvLSTM network. The ConvLSTM structure replaces the inner product of LSTM from the Hadamard product with convolution. It processes hidden and input states in two dimensions making it able to capture spatio-temporal motion patterns. The primary benefit of using this structure is to capture the adjacent spatio-temporal relationships and identify a saliency map of the inter-vehicle interaction based on risk, which can improve the overall accuracy of trajectory prediction. The input is the

Risk Map sequence passing through the multiple ConvLSTM layers which have convolutional structures in both input-to-state and state-to-state transitions [21]. These structures are represented in the following equations,

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \quad (6)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \quad (7)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \quad (8)$$

$$O_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \quad (9)$$

$$H_t = O_t \circ \tanh(C_t) \quad (10)$$

where  $\sigma$  is the logistic sigmoid function,  $i_t$ ,  $f_t$ ,  $C_t$ ,  $O_t$  and  $H_t$  are the input gate, forget gate, cell activation, output gate, and cell output at time  $t$ , respectively. '\*' denotes the convolution operator and 'o' denotes the Hadamard product. In Equation (6),  $W$  represents the filter matrices which connect different gates and  $b$  is the corresponding bias vector. We have added padding before applying the convolution operation to ensure that the states have the same number of rows and same number of columns as the inputs. The ConvLSTM learns to leverage the temporal features during training and compresses the whole input sequence into a hidden state tensor. To obtain the affine transformation parameter, a fully connected layer with a Rectified linear units (ReLU) activation function is used. These parameters are concatenated with the encoder  $c_i$  and then is fed to the decoder for further processing. The decoder unfolds this hidden state to predict the vehicle trajectory. The context vector contains a representation of input sequences in the encoder and RM which carries the most significant spatio-temporal features from the input sequence. The context vector  $c_i^{dec}$  is calculated and passed to the decoder as their initial cell states.

#### C. DECODER LSTM

The decoder module reads the encoded sequences and makes a multi-step prediction in the output sequence. The decoding function,  $f$ , takes the context vector,  $c_i^{dec}$ , at time  $i$ , the hidden state and output of the decoder at time  $i - 1$  shown by  $S_{i-1}$ , and  $y_{i-1}$ , respectively, and outputs the predicted trajectory distribution which is given by,

$$y_i = f(s_{i-1}, y_{i-1}, c_i^{dec}) \quad (11)$$

#### D. INPUT

The proposed framework takes two channels of information as input to the system; trajectory information through the Occupancy Map (OM) and risk associated with the traffic scene through the Risk Map (RM). In the following paragraph, we describe each channel in detail.

The first channel of information is encoder input. The encoder takes track history of the ego-vehicle and all the surrounding vehicles through an Occupancy Map (OM). To define the OM, we surround the ego-vehicle with a stationary grid, where each column corresponds to a single lane, and the rows are separated by a length of approximately one standard vehicle. Vehicles in a traffic scene, adjust their trajectory

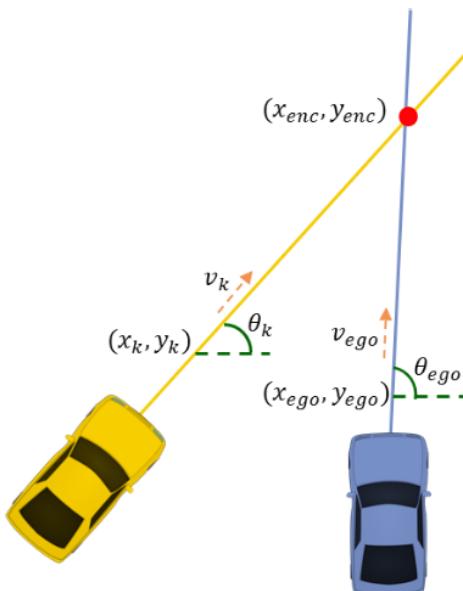
with the surrounding vehicle trajectories generating interaction amongst them. The impact of this interaction between vehicles is influenced by the relative distance and velocity between them. In this study, we utilise the OM to define the *weight vector*  $W_v$  reflecting the impact of surrounding vehicle trajectories on the ego-vehicle. The  $W_v$  can be calculated by,

$$\text{weights}_{i,j} \propto f(\text{Distance}_{i,j}, \text{Velocity}_{i,j}) \quad (12)$$

$$W_{v,i}^k \propto \frac{v_{r,i}^k}{d_{r,i}^k} \quad \forall k \in \{1, 2, \dots, 39\} \quad (13)$$

where  $d_{r,i}$  and  $V_{r,i}$  are the relative distance and relative velocity between the  $k^{th}$  vehicle and the ego-vehicle at the  $i^{th}$  time step, respectively.

The second channel of information embedded in the proposed framework is the Risk Map which is an auxiliary input to generate the context vector for the decoder module. This input allows the system to automatically extract the risk pattern associated with the traffic scene in each time step  $i$ . In contrast to existing approaches [2] which consider a simplified calculation of the safety factor (i.e., only longitudinal velocity is considered), this paper presents a comprehensive definition of TTC where we consider both lateral and longitudinal velocity to describe the safety factor. The method of how TTC is calculated for both lateral and longitudinal velocity is depicted in Figure 2.



**FIGURE 2.** TTC calculation using both lateral and longitudinal velocity [22].

Here,  $(x_{enc}, y_{enc})$  is the encounter point which is where a collision occurs if vehicles continue with the same velocity and direction [22], it is given by:

$$x_{enc} = \frac{(y_{ego} - y_k)}{\tan(\theta_k) - \tan(\theta_{ego})} - \frac{x_{ego} \cdot \tan(\theta_{ego}) - x_k \cdot \tan(\theta_k)}{\tan(\theta_k) - \tan(\theta_{ego})} \quad (14)$$

$$y_{enc} = \frac{(x_{ego} - x_k)}{\cot(\theta_k) - \cot(\theta_{ego})} - \frac{y_{ego} \cdot \cot(\theta_{ego}) - y_k \cdot \cot(\theta_k)}{\cot(\theta_k) - \cot(\theta_{ego})} \quad (15)$$

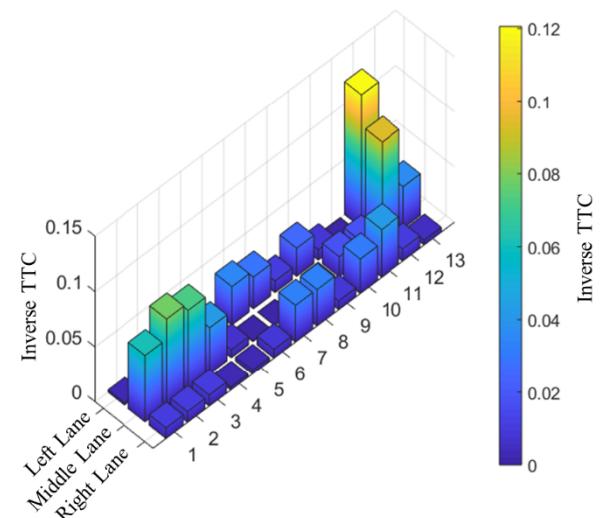
where  $(x_{ego}, y_{ego})$  and  $(x_k, y_k)$  are the coordinates of the ego-vehicle and the  $k^{th}$  surrounding vehicle in meters, respectively. The  $\theta[\text{Rad}]$  is the heading of the vehicle which is obtained from the angle between lateral velocity,  $v_y$ , and longitudinal velocity,  $v_x$  ( $\theta = \arctan \frac{v_y}{v_x}$ ). If vehicle  $j$  continues its path with the same heading,  $t_{enc,j}$  is the time taken to reach the encounter point.

$$t_{enc,j} = \frac{\sqrt{(x_j - x_{imp})^2 + (y_j - y_{imp})^2}}{\sqrt{v_{x,j} + v_{y,j}}} \quad (16)$$

The TTC between the ego-vehicle and the  $k^{th}$  surrounding vehicle can be obtained from  $t_{enc,ego} - t_{enc,k}$ .

$$|TTC = t_{enc,ego} - t_{enc,k}| \leq \epsilon \quad (17)$$

Here,  $\epsilon$  is a safety margin which defines the degree of hazard for the situation. In  $\epsilon = 0$ , both vehicles reach the encounter point at the same time (i.e., means a collision). The higher  $\epsilon$  value shows that the safety factor is more conservative (i.e., higher  $\epsilon$  value denotes lower risk of collision). The literature shows that a TTC of more than 6 seconds is considered a safe distance between two vehicles [23]; however, we will discuss the impact of TTC values greater than 6 seconds on the accuracy of trajectory prediction in the simulation result section. Figure 3, shows the average of the TTC inverse value in the NGSIM dataset.



**FIGURE 3.** Average of inverse TTC values in NGSIM dataset.

#### IV. SIMULATION RESULTS AND DISCUSSION

The following section describes the datasets, evaluation metrics, and the model performance of the trajectory prediction framework. We utilised LSTM with 128 units for the encoder and decoder module with a batch size of 32. Models are trained using the Adam optimizer [24] with a learning rate of 0.001 and ReLU activation with  $\alpha = 0.1$ . The loss function of the training process adopts MSE and we set 20% dropout in the training process. Two ConvLSTM layers are stacked,

**TABLE 1.** An overview of different trajectory types in datasets.

Trajectories	NGSIM			HighD		
	Train	Val.	Test	Train	Val.	Test
Total	8695	970	1885	68132	9422	19466
8 seconds windows	89632	15082	335227	1948158	2603 40	507567
Straight	83232	14285	334206	1943541	2598 04	506271
Lane changes	6400	797	1021	4617	536	1296

where each layer has 128 hidden states with  $2 \times 2$  kernels. The decoder consists of the 128 LSTM cell followed by an FC layer.

#### A. DATASETS AND DATA PREPARATION

To train and evaluate the proposed framework, we first need a detailed dataset covering a variety of vehicle trajectories. We extract all the eligible trajectories from two naturalistic vehicle trajectory benchmark datasets, NGSIM US-101 and I-80 and HighD. The NGSIM US-101 and I-80 dataset [25], [26] consist of real highway driving scenarios captured using multiple overhead cameras observing sections of highway in the US in 2005 at 10 Hz. It is a large size dataset with dense traffic flow which has been widely used in the literature especially for trajectory prediction [2], [4], [11].

The second dataset is the HighD dataset [27] which was captured by a camera-equipped drone in 2017 and 2018 at 25Hz. The dataset is collected from an aerial perspective of six different highways in Germany including 5600 complete lane changing trajectories.

NGSIM and HighD datasets have different proportions of lane changing and straight through trajectories. Table 1, overviews the different trajectory types in both datasets. We randomly divided each of the datasets into training, validating, and testing sets where the training set contains 70% of the trajectories, and the validating and testing sets contain the remaining 10% and 20%, respectively. In each dataset, we consider one of the vehicles as the ego-vehicle and the other vehicles as surrounding vehicles. In the proposed model, we observe 3 seconds of the track history of all vehicles in the traffic scene to predict the next 5 seconds of the ego-vehicle trajectory, therefore each trajectory segment needs to be 8 seconds long. In order to reduce model complexity, we have down sampled the 8 second trajectory with a time step  $\Delta t$  equal to 0.2 seconds (5 Hz). The down-sampling process provides a sequence with a length of 40 steps. We have utilised the first 15 steps (i.e. 3 seconds) as the track history sequence and the rest of the 25 steps (i.e. 5 seconds) as the output sequence. We also transformed the initial values of each trajectory ( $x = 0, y = 0$ ) making the model independent of road geometry. The independence of trajectory prediction from road geometry makes the model more flexible allowing it to be employed across different environments.

#### B. EVALUATION METRICS

We report the trajectory prediction accuracy with the *Root of the Mean Square Error (RMSE)* metric since it has been widely used in previous work [2], [11], [17], [28]. The RMSE is a quadratic scoring rule measuring the average magnitude of the displacement errors of predicted positions and real positions during the prediction time horizon. This metric gives relatively high weight to large errors which penalises them and can be defined as follow,

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{i,t}^{\text{pred}} - x_{i,t}^{\text{obs}})^2} \quad (18)$$

where  $n$  is the number of trajectories in the testing set,  $x_{i,t}^{\text{pred}}$  is the predicted position and  $x_{i,t}^{\text{obs}}$  is the respective observed positions for the trajectory  $i$  at  $t^{\text{th}}$  time instance.

#### C. ANALYZING MODEL PERFORMANCE

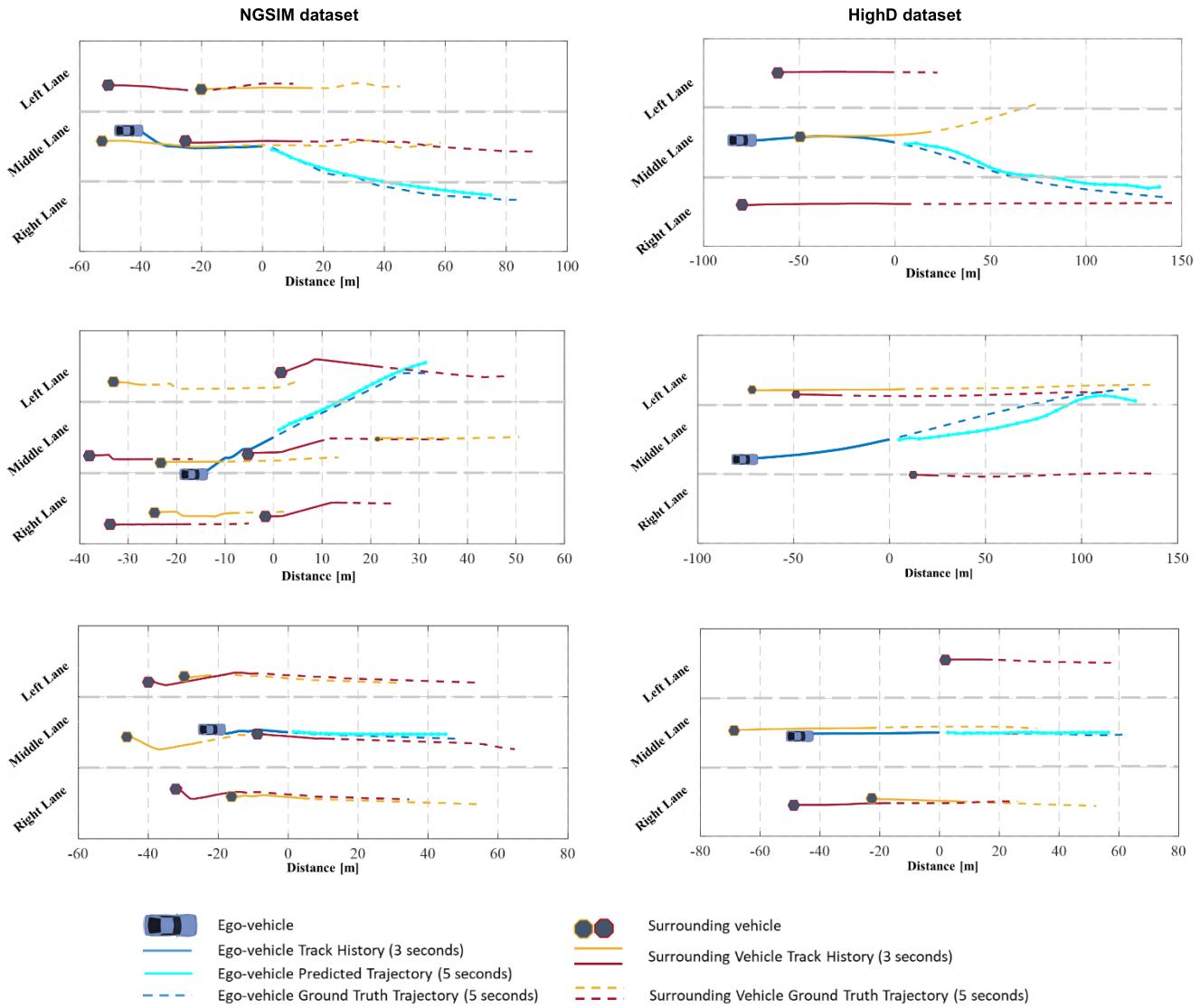
In this section, we analyse the performance of the proposed trajectory prediction model in both quantitative and qualitative aspects to present the experimental results.

To intuitively show the trajectory prediction outcomes, three sample trajectories including change lane right (first row), change lane left (second row), and going straight (third row) are randomly selected from both the NGSIM and HighD datasets, see Figure 4. The first row shows two examples of lane change where the ego-vehicle performs two types of interaction for the same trajectory type. One scenario with a vehicle in the destination lane and one change lane to a free lane. In Figure 4, the surrounding vehicles' 3 second track history (solid line) and ground truth trajectory (dash line) are shown in red and yellow, the track history (solid line) and ground truth (dash line) trajectory for the ego-vehicle are in navy blue, and the five second predicted trajectory for the ego-vehicle is plotted in light blue. It can be clearly seen that the proposed approach achieved a promising result in predicting different trajectory types while it considered the interaction with surrounding vehicles.

#### 1) WEIGHT VECTOR AND RISK MAP IMPACT

In the experimental results depicted in Figure 5, we investigate the impact of considering a *weight vector* ( $W_v$ ) and Risk Map on the accuracy of the proposed system. To evaluate the effectiveness of these factors, we present the RMSE performance of the proposed model for lateral and longitudinal trajectory prediction with three modifications.

In one experiment, we trained the system only with kinematic parameters, in the second, we embed  $W_v$  as well as kinematic parameters, and finally in the third experiment, we utilised the Risk Map,  $W_v$ , and kinematic parameters while the other attributes of the three models are unchanged. The goal is to check the accuracy of the trajectory prediction for different time horizons. Figure 5 shows that, in both datasets, systems utilising  $W_v$  and the Risk Map decrease the RMSE value for both lateral and longitudinal trajectory.



It should be noted that this improvement is significantly larger in lateral trajectory prediction showing that  $W_v$  and the Risk Map provide the system with better lateral trajectory insight compared with solely using Kinematic parameters. In longitudinal trajectory prediction, embedding the Risk Map and  $W_v$  results in a slight improvement compared to the accuracy of the system which is trained with  $W_v$  and Kinematic parameters. As an example, in the first row of Figure 5, the RMSE value of the model is decreased by adding the Risk Map (i.e., blue line), however this reduction is more noticeable for lateral trajectory prediction.

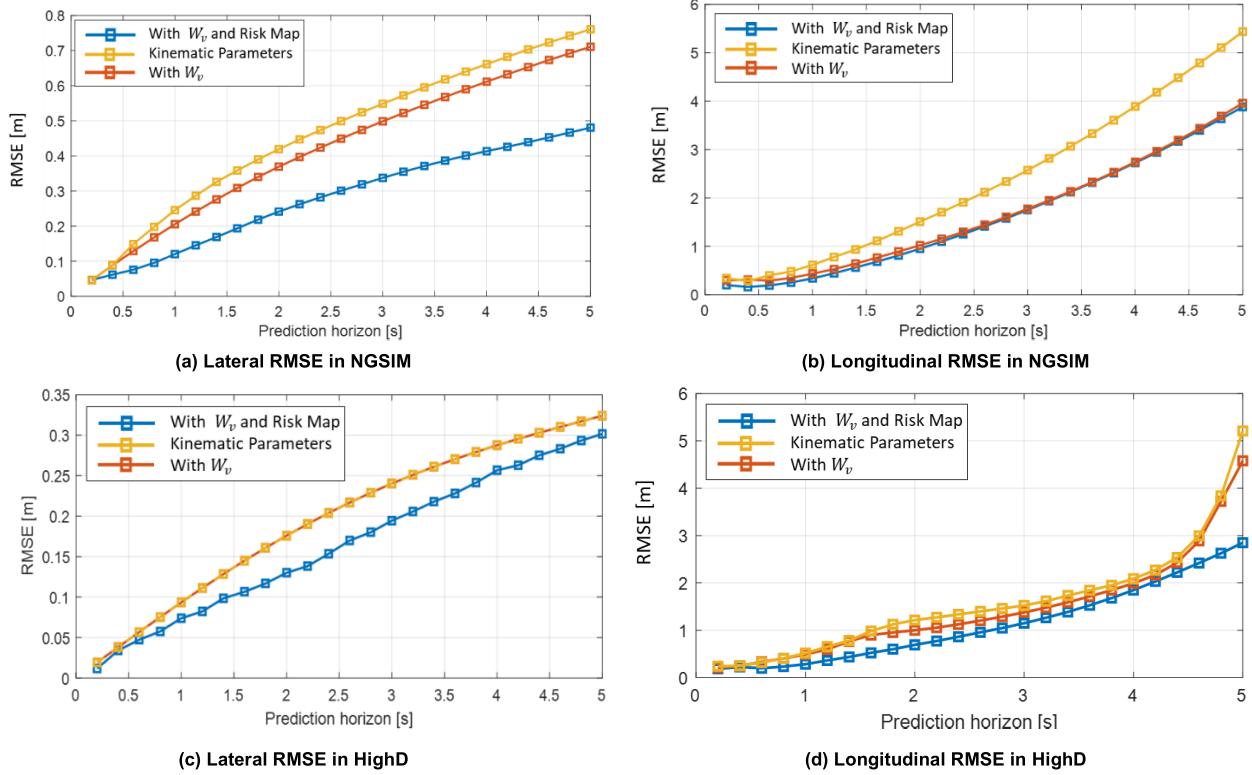
## 2) COMPARISON WITH STATE-OF-THE-ART

To evaluate the proposed approach, we pursue a direct comparison with state-of-the-art vehicle trajectory prediction using the same dataset (i.e., NGSIM). The results show that

the proposed method outperforms the state-of-the-art model and decreases the overall RMSE value of the system by 10 percent on average.

Table 2, summarises the RMSE values comparing the proposed methods with the baseline trajectory prediction models in the literature [4], [5], [11], [28]. The following baselines are evaluated:

- i. **Vanilla LSTM (V-LSTM):** This method utilises the track history of the ego-vehicle in the encoder LSTM and produces a unimodal output distribution with the LSTM decoder.
- ii. **Manoeuvre-LSTM (M-LSTM) [11]:** This model employs an encoder-decoder framework where the ego-vehicle and surrounding vehicle trajectories are fed into the encoder. The decoder uses an encoded vector as well as manoeuvre encoding and generates a multi-modal trajectory prediction.



**FIGURE 5.** Effectiveness of considering  $W_v$  and Risk Map in the vehicle trajectory prediction.

**TABLE 2.** RMSE comparison of the proposed method with the baseline models and state-of-the-art.

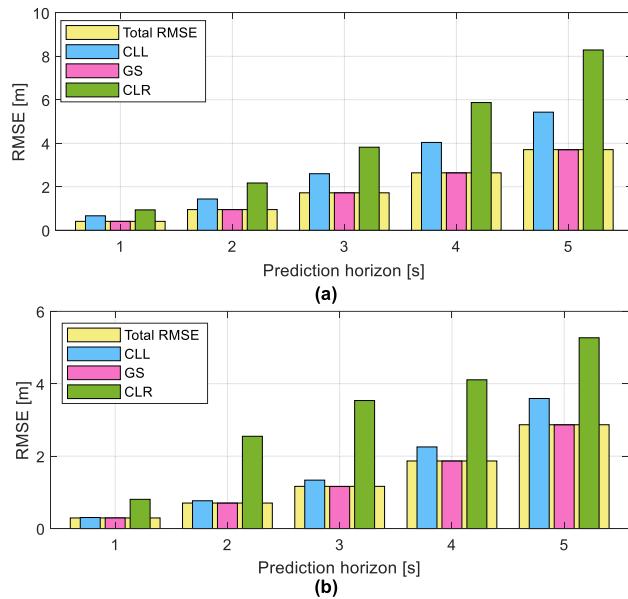
Datasets	Prediction horizon (s)	V-LSTM	M-LSTM [11]	S-LSTM [29]	CS-LSTM [4]	NLS-LSTM [5]	Proposed method
NGSIM	1	0.68	0.58	0.65	0.61	0.56	<b>0.41</b>
	2	1.65	1.26	1.31	1.27	1.22	<b>0.95</b>
	3	2.91	2.12	2.16	2.09	2.02	<b>1.72</b>
	4	4.46	3.24	3.25	3.10	3.03	<b>2.64</b>
	5	6.27	4.66	4.55	4.37	4.30	<b>3.87</b>
HighD	1	N/A	N/A	0.22	0.22	0.22	<b>0.22</b>
	2	N/A	N/A	0.62	0.61	0.61	<b>0.61</b>
	3	N/A	N/A	1.27	1.24	1.24	<b>1.16</b>
	4	N/A	N/A	2.15	2.10	2.10	<b>1.8</b>
	5	N/A	N/A	3.41	3.27	3.27	<b>2.8</b>

- iii. **LSTM with fully connected social pooling (S-LSTM) [28]:** This model utilises fully connected social pooling to generate a unimodal output distribution.
- iv. **Convolutional Social Pooling (CS-LSTM) [4]:** This model utilises convolutional social pooling and generates a unimodal output distribution.
- v. **Non-local Social Pooling (NLS-LSTM) [5]:** This is an LSTM encoder-decoder model using a social pooling mechanism that combines local and non-local operations.

The comparison results show that the proposed DLM structure has better performance with  $\sim 11\%$  and  $\sim 14\%$  improvement in RMSE for NGSIM and HighD, respectively. As depicted in Table 2, the RMSE value for the system trained with the NGSIM dataset is higher than the system trained with the HighD dataset. The reason for this difference has also been noted in previous investigations [5] and is likely related to dataset size differences or is due to annotation inaccuracies resulting in physically unrealistic vehicle behaviours in the NGSIM dataset [29].

### 3) ANALYSING THE ALGORITHM PERFORMANCE WITH INSIGHT INTO THE DATA

Up until this point, we have reported and compared the accuracy of our proposed method with existing approaches. In this part, we aim to analyse the algorithm performance with more insight into the data. Figure 6 shows the plot of RMSE values for different lane changes (left/right) and straight trajectories as well as the average RMSE of the system.



**FIGURE 6.** RMSE value of the proposed model for different trajectories (a) NGSIM, and (b) HighD dataset.

The bar chart demonstrates that change lane right and change lane left have a higher RMSE compared with go straight trajectories.

In other words, the system has better performance predicting go straight trajectories than turn trajectories. The available datasets contain a higher proportion of go straight trajectories however most errors happen for lane change trajectories. By reporting the average error of the system, errors in lane change prediction may be hidden since the system can accurately predict go straight trajectories. Therefore, to more accurately evaluate and express the representation of model performance, we report the RMSE value of the proposed model for different types of trajectories for a 5 second prediction horizon in both datasets.

Figure 6, shows the prediction error for different trajectories including Change Lane Left (CLL), Go Straight (GS), and Change Lane Right (CLR). It is observed that the error of GS is lower than the other trajectories. This result is expected since lane keeping (Go Straight) trajectories are less complex to predict compared with scenarios where the ego-vehicle intends to change lane. Figure 6 also shows that the RMSE value caused by CLR is more than CLL trajectories indicating the trained system is less confident in predicting CLR trajectories. This can be the result of the lower number

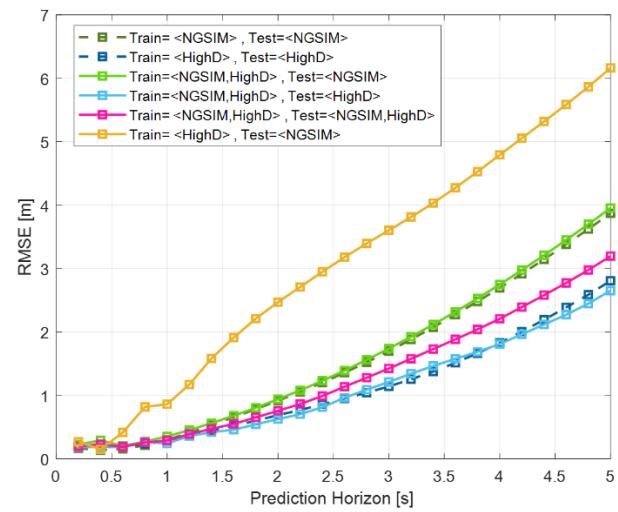
of CLR trajectories compared with CLL and GS trajectories existing in datasets.

### 4) ANALYSING THE INDEPENDENCE OF THE SYSTEM TO A SPECIFIC ENVIRONMENT

To illustrate the independence of the system to a specific environment, we train and test the DLM structure with different combinations of both datasets for training and testing, such as:

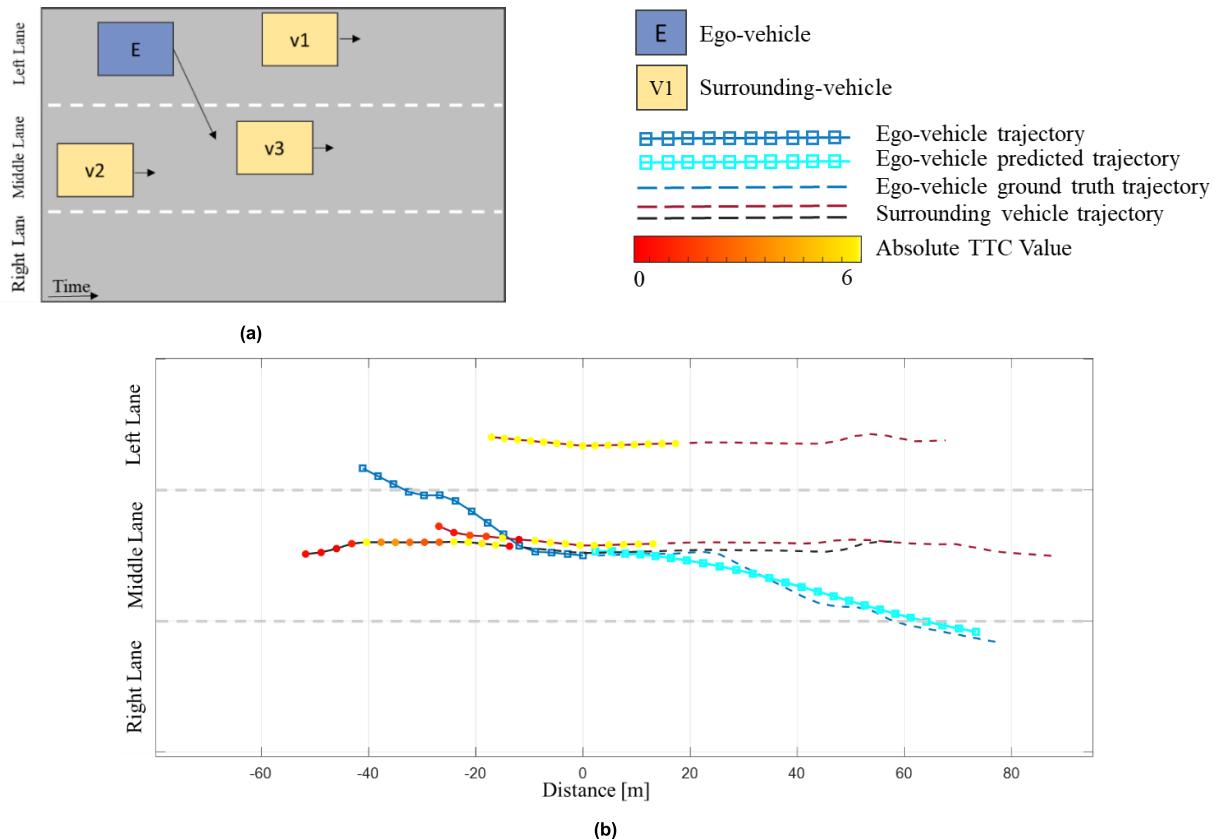
- a) Train and test with the same dataset
- b) Train with both datasets and test with each of the datasets, separately
- c) Train with a combination of both datasets and test with both datasets
- d) Train with the HighD dataset and test only with the NGSIM dataset (Cross dataset prediction).

The result, Figure 7, shows that the system maintains its performance without the tendency to overfit to one of the datasets. Thus, combined training across multiple datasets does not substantially decrease accuracy of prediction within each individual dataset. In the cross dataset experiment, where we trained the system with the HighD dataset and tested it with the NGSIM dataset, the prediction result is less accurate implying that cross learning is not a feature of this approach due to uniqueness of each dataset. However, the training across multiple datasets does not negatively influence the accuracy of individual dataset prediction.



**FIGURE 7.** The RMSE value of the proposed system using different training datasets.

In order to see the impact of the Risk Map on the DLM structure, we attempt to gain a better understanding of our model performance in the qualitative analysis. We investigate the effect of the surrounding vehicle's trajectories on the ego-vehicle trajectory prediction with respect to the TTC described in Equations (10) and (11). In Figure 8, a trajectory example is presented where the ego-vehicle is interacting with three vehicles in the traffic scene, see Figure 8(a).



**FIGURE 8.** An illustration of dynamic vehicle interaction and the effect on risk/TTC.

The ego-vehicle track history starts at the left lane where  $V_1$  is going straight and finishes at the middle lane where there are two vehicles which are going straight. Different TTC values are presented with dot point' colours in the track history of the surrounding vehicles. The TTC pattern changes from  $|TTC| > 6$  (i.e., yellow scattered points) to smaller TTC values showing the dynamic interaction, and hence risk, between vehicles in the traffic scene, see Figure 8 (b). When the ego-vehicle starts changing lane, the TTC values for  $V_2$  and  $V_3$  become red meaning that risk has increased in that lane. As can be seen in Figure 8 (b), the ego-vehicle begins the lane change manoeuvre but seems to straighten midway, possibly due to the recognition of the trailing vehicle in the new lane. This change in approach is likely due to the driver's re-evaluation of risk as they recognise changes in the vehicle environment. When the ego-vehicle starts to change lane, TTC values for both  $V_2$  and  $V_3$  decrease again (i.e., orange scattered points). The relative risk relationships change due to behaviours of both the ego vehicle and surrounding vehicles, illustrated by differences in levels of risk for  $V_2$  and  $V_3$  over time. Different combinations of behaviours may lead to a similar risk outcome. As equilibrium is achieved between vehicles, risk once again becomes yellow. Lane changing behaviour is influenced by risk recognition by drivers and thus influences the precision of lane change prediction.

Ultimately, the result of this investigation shows that the deep feature selection of the Risk Map using ConvLSTM helps to create an accurate trajectory prediction and captures the dynamic interaction of the ego-vehicle with other vehicles in the traffic scene.

## V. CONCLUSION

In this paper, we have presented a novel end-to-end Dual Learning Model (DLM) for vehicle trajectory prediction. We consider all traffic scene interactions including their relative distance and velocity to the ego-vehicle and apply a comprehensive safety factor definition based on TTC which results in precise trajectory prediction. Two channels of information are embedded within the proposed framework. In the second information channel, we used ConvLSTM, wherein the inner product of LSTMs is replaced with convolution to maintain spatio-temporal motion patterns. We added more expressive features to propose a precise trajectory prediction framework without increasing complexity. We investigate the effectiveness of our proposed system by testing and evaluating it using two distinct datasets, NGSIM (i.e. I-80, US-101) and HighD, resulting in  $\sim 11\%$  and  $\sim 14\%$  improvement in RMSE values of prediction for NGSIM and HighD datasets, respectively. Additionally, we would like to point out that this is the first time that the risk associated with a traffic scene is considered in a trajectory prediction

framework. It allows the prediction model to learn different patterns of influence from surrounding vehicles which results in more precise trajectory prediction compared with baseline models where they just consider the ego-vehicle and surrounding vehicle kinematics parameters.

We acknowledge the limitation of considering other road environments such as curved roads or roundabouts at this stage of our research. The future direction of this research is to investigate the effectiveness of the proposed method for different scenarios such as roundabouts and intersections. In addition, investigating the influence of Cooperative ITS (C-ITS), the effect of using communication information in the proposed dual learning Model for decision making in automated vehicles, is a worth future investigation.

## REFERENCES

- [1] *Global Status Report on Road Safety*, World Health Org., Geneva, Switzerland, 2018.
- [2] S. Dai, L. Li, and Z. Li, “Modeling vehicle interactions via modified LSTM models for trajectory prediction,” *IEEE Access*, vol. 7, pp. 38287–38296, 2019.
- [3] N. Deo, A. Rangesh, and M. M. Trivedi, “How would surround vehicles move? A unified framework for maneuver classification and motion prediction,” *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 129–140, Jun. 2018.
- [4] N. Deo and M. M. Trivedi, “Convolutional social pooling for vehicle trajectory prediction,” 2018, *arXiv:1805.06771*. [Online]. Available: <https://arxiv.org/abs/1805.06771>
- [5] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, “Non-local social pooling for vehicle trajectory prediction,” in *Proc. 4th IEEE Intell. Vehicles Symp.*, Paris, France, 2019, pp. 975–980, doi: [10.1109/IVS.2019.8813829](https://doi.org/10.1109/IVS.2019.8813829).
- [6] S. Lefèvre, D. Vasquez, and C. Laugier, “A survey on motion prediction and risk assessment for intelligent vehicles,” *Robomech J.*, vol. 1, p. 1, Jul. 2014.
- [7] M. Schreier, V. Willert, and J. Adamy, “Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems,” in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 334–341.
- [8] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, “Vehicle trajectory prediction based on motion model and maneuver recognition,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 4363–4369.
- [9] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Y. Yong, J. Yoder, C. Tay, K. Mekhnacha, and A. Negre, “Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety,” *IEEE Intell. Transp. Syst. Mag.*, vol. 3, no. 4, pp. 4–19, Oct. 2011.
- [10] J. Schlechtriemen, F. Wirthmueller, A. Wedel, G. Breuel, and K.-D. Kuhnert, “When will it change the lane? A probabilistic regression approach for rarely occurring events,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 1373–1379.
- [11] N. Deo and M. M. Trivedi, “Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1179–1184.
- [12] S. Patel, B. Griffin, K. Kusano, and J. J. Corso, “Predicting future lane changes of other highway vehicles using RNN-based deep models,” 2018. *arXiv:1801.04340*. [Online]. Available: <https://arxiv.org/abs/1801.04340>
- [13] Q. Tran and J. Fir, “Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression,” in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 918–923.
- [14] M. Brand, N. Oliver, and A. Pentland, “Coupled hidden Markov models for complex action recognition,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 97, Nov. 2002, p. 994.
- [15] N. Oliver and A. Pentland, “Graphical models for driver behavior recognition in a SmartCar,” in *Proc. IEEE Intell. Vehicles Symp.*, Nov. 2002, pp. 7–12.
- [16] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr, “A combined model-and learning-based framework for interaction-aware Maneuver prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1538–1550, Jun. 2016.
- [17] S. 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, *arXiv:1802.06338*. [Online]. Available: <https://arxiv.org/abs/1802.06338>
- [18] A. Milstein, “Occupancy grid maps for localization and mapping,” in *Motion Planning*, Rijeka, Croatia: InTech, 2008.
- [19] X. Yan, Y. Zhang, and L. Ma, “The influence of in-vehicle speech warning timing on drivers’ collision avoidance performance at signalized intersections,” *Transp. Res. C, Emerg. Technol.*, vol. 51, pp. 231–242, Feb. 2015.
- [20] W. Zhan, A. De La Fortelle, Y.-T. Chen, C.-Y. Chan, and M. Tomizuka, “Probabilistic prediction from planning perspective: Problem formulation, representation simplification and evaluation metric,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1150–1156.
- [21] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.
- [22] R. Miller and Q. Huang, “An adaptive peer-to-peer collision warning system,” in *Proc. Veh. Technol. Conf. IEEE 55th Veh. Technol. Conf. VTC Spring*, vol. 1, Jun. 2003, pp. 317–321.
- [23] S. G. Dehkordi, G. S. Larue, M. E. Cholette, A. Rakotonirainy, and H. A. Rakha, “Ecological and safe driving: A model predictive control approach considering spatial and temporal constraints,” *Transp. Res. D, Transp. Environ.*, vol. 67, pp. 208–222, Feb. 2019.
- [24] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [25] J. Colyar and J. Halkias, “US highway i-80 dataset,” Federal Highway Admin. (FHWA), Washington, DC, USA, Tech. Rep. FHWA-HRT-06-137, 2006.
- [26] J. Colyar and J. Halkias, “US highway 101 dataset,” Federal Highway Admin. (FHWA), Washington, DC, USA, Tech. Rep. FHWA-HRT-07-030, 2007.
- [27] 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 *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2118–2125.
- [28] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human trajectory prediction in crowded spaces,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.
- [29] B. Coifman and L. Li, “A critical evaluation of the next generation simulation (NGSIM) vehicle trajectory dataset,” *Transp. Res. B, Methodol.*, vol. 105, pp. 362–377, Nov. 2017.



**MAHROKH KHAKZAR** received the B.S. and M.S. degrees in electrical engineering from the Azad University of Najafabad, Iran, in 2011 and 2015, respectively. She is currently pursuing the Ph.D. degree with the Queensland University of Technology, Brisbane, QLD, Australia. From 2016 to 2018, she was a Research Assistant with the Centre for Accident Research and Road Safety-Queensland (CARRSQ), where she has conducted research on different multidisciplinary projects with a focus on big data analysis, artificial intelligence, machine learning, and image and video processing. Her research interests include machine learning, intelligent transportation systems, intelligent vehicle, and naturalistic data analysis.



**ANDRY RAKOTONIRAINY** is currently the Director of the Centre for Accident Research Road Safety-Queensland (CARRS-Q) and a Founder of its Intelligent Transport Systems Human Factors Research Program, CARRS-Q. He has 20 years of research and management experience in computer science and brings advanced expertise in road safety and ITS. He has established the CARRS-Q Advanced Driving Simulator Laboratory, which was the most advanced driving simulator facility in Australia. He has been proactive in investigating the use of existing and emerging ITS from multiple disciplines. It incorporates disciplines such as computer science, mathematics, human factors, engineering, psychology, and sociology. His research has made extensive use of driving simulators, traffic simulators, and instrumented vehicles for developing system prototypes, assessing cost benefits, understanding human errors, and evaluating system deployment. His research on ITS has received numerous competitive grants and generated extensive interest from road safety stakeholders.



**SEPEHR G. DEHKORDI** received the B.S. degree in electrical engineering from the Azad University of Najafabad, in 2011, the M.S. degree in control and automation engineering from Universiti Putra Malaysia, in 2015, and the Ph.D. degree from the Queensland University of Technology, in 2019. He is currently a Research Associate with the Centre for Accident Research Road Safety-Queensland (CARRS-Q), researching on the connected automated vehicle and intelligence transportation system (ITS). His research interests and expertise cover several areas including optimization techniques in control theory, intelligent control, road safety approaches in ITS, ecological driving behaviors, system identification, and dynamic system modeling.

• • •



**ANDY BOND** received the Ph.D. degree in computer science from the Victoria University of Wellington, which investigated the application of artificial intelligence methods to the scheduling of tasks across a distributed system of computing resources, learning from and predicting behavior and optimization. He spent a decade in digital health with the Australian Government working on electronic health records and health informatics. He has played a leading role in digital health standards, in particular in the application of national and international digital health architecture and interoperability. His early career was in research and development culminating in the position of the Chief Scientist with the Distributed Systems Technology Centre (DSTC), where he was responsible for client engagement with research and the transition of technology into commercial opportunities. He is currently the Director of future mobility with the Queensland University of Technology (QUT). His role looks to the future of transport through the perspective of new technology opportunities and patterns of use based on work in road safety, law, and robotics.