Vehicle Trajectory Prediction Using LSTMs With Spatial–Temporal Attention Mechanisms

©SHUTTERSTOCK.COM/UNIKYLUCKK

Lei Lin
Is with the Goergen Institute for Data Science, University of Rochester
Rochester, New York, 14627, USA. Email: lei.lin@rochester.edu

Weizi Li*
Is with the University of Memphis, Memphis, Tennessee, 38152, USA.
Email: wli@memphis.edu

Huikun Bi
Is with the Institute of Computing Technology, Chinese Academy of Sciences
Beijing, 100190, China. Email: bihuikun@ict.ac.cn

Lingqiao Qin
Is with the University of Wisconsin–Madison, Madison, Wisconsin, 53704, USA.
Email: Lingqiao.qin@wisc.edu

Digital Object Identifier 10.1109/MITS.2021.3049404
Date of current version: 8 February 2021

*Corresponding authors

Abstract—Accurate vehicle trajectory prediction can benefit a variety of intelligent transportation system applications ranging from traffic simulations to driver assistance. The need for this ability is pronounced with the emergence of autonomous vehicles as they require the prediction of nearby vehicles' trajectories to navigate safely and efficiently. Recent studies based on deep learning have greatly improved prediction accuracy. However, one prominent issue of these models is the lack of model explainability. We alleviate this issue by proposing spatiotemporal attention long short-term memory (STA-LSTM), an LSTM model with spatial-temporal attention mechanisms for explainability in vehicle trajectory prediction. STA-LSTM not only achieves comparable prediction performance against other state-of-the-art models but, more importantly, explains the influence of historical trajectories and neighboring vehicles on the target vehicle. We provide in-depth analyses of the learned spatial–temporal attention weights in various highway scenarios based on different vehicle and environment factors, including target vehicle class, target vehicle location, and traffic density. A demonstration illustrating that STA-LSTM can capture and explain fine-grained lane-changing behaviors is also provided. The data and implementation of STA-LSTM can be found at https://github.com/leilin-research/VTP.

Since the first autonomous driving competition hosted by DARPA in 2005 [1], autonomous vehicles (AVs) have attracted extensive attention from both academia and industry. With the recent advancements in sensing as well as machine learning, the research and development of autonomous driving have achieved tremendous progress.

There are two main approaches to achieve autonomous driving. The first is the end-to-end approach that directly maps raw sensor data to control commands using a single model—commonly one or more neural networks [2]–[5]. The second is the traditional engineering approach [6], [7] that involves multiple modules such as detection, tracking, prediction, and planning. While both approaches have merits and drawbacks—as the safety of AVs is the leading concern—the traditional engineering approach is likely to prevail in the near future because of its better model interpretability and controllability.

One crucial task of the traditional engineering approach is to predict the trajectories of vehicles surrounding the AV—information that is required to achieve safe and robust driving. In this article, we focus on vehicle trajectory prediction on highways, where the dominant traffic participants are cars and trucks. We refer to the vehicle whose trajectory is being predicted as the target vehicle and the surrounding vehicles of the target vehicle as neighboring vehicles.

Among the many techniques for predicting vehicle trajectories, recurrent neural networks (RNNs) have offered state-of-the-art performance [8]–[12]. RNNs take the historical trajectory data of the target vehicle as the input and predict its trajectory over a certain time horizon. RNNs are particularly effective because they consider both the local information among vehicles (e.g., instantaneous interactions between the target vehicle and its leading vehicle) and the long-term dependencies stored in memory cells [8], [9].

While RNNs are effective in prediction, they offer limited model explainability. In particular, how the long-term information embedded in historical trajectories [8], [9] and the information of neighboring vehicles [11], [12] impact prediction is left unexplored. In this article, we aim to answer the following questions: Which part of the historical trajectories of the target vehicle or neighboring vehicles determines the future motion of the target vehicle? Which neighboring vehicles influence the target vehicle more? Where would these neighboring vehicles be? Answering these questions from the temporal–spatial perspective can help us better understand a driver's decision-making process, identify various driving styles, design realistic traffic simulation models, and ultimately assist in developing safe and efficient autonomous driving.

In an effort to address these questions, we propose the spatiotemporal attention long short-term memory (STA-LSTM), an LSTM model with spatial–temporal attention mechanisms for explainability in vehicle trajectory prediction. STA-LSTM not only achieves comparable prediction accuracy to other state-of-the-art techniques but also explains the influence of historical trajectories and neighboring vehicles on the target vehicle via attention weights. STA-LSTM is learned and evaluated using the Next Generation Simulation (NGSIM) data set [13].

We provide in-depth analyses of the learned attention weights in scenarios that contain different sets of vehicles and environmental factors, including target vehicle classes (e.g., cars and trucks), target vehicle locations, and neighboring vehicle densities. We also analyze the attention weights associated with specific driving behaviors of the target vehicle and find that the learned attention weights can be used to interpret the target vehicle's lane-changing behaviors. The data and implementation of STA-LSTM can be found at https://github.com/leilin-research/VTP. In summary, the main contributions of this work are as follows:

- STA-LSTM, an LSTM model with spatial–temporal attention mechanisms, is developed for predicting vehicle trajectories.
- The proposed attention mechanisms at the temporal level can identify important historical trajectories for determining future behaviors of the target vehicle.
- The proposed attention mechanisms at the spatial level can rank neighboring vehicles in terms of their influences on the target vehicle.
- In-depth analyses of the learned attention weights in traffic scenarios with various vehicle and environment factors are provided.
- Specific driving behaviors of the target vehicle through the learned attention weights are analyzed. In particular, lane-changing behaviors of the target vehicle are found to be explainable through the attention weights.

> One crucial task of the traditional engineering approach is to predict the trajectories of vehicles surrounding the AV—information that is required to achieve safe and robust driving.

Related Work

Vehicle Trajectory Prediction Using Traditional Methods

Conventionally, three types of approaches exist for vehicle trajectory prediction: physics-based, maneuver-based, and interaction-aware [14]. Physics-based methods usually consider vehicle kinematic and dynamic constraints, such as yaw rate and acceleration rate, and environmental factors, such as the friction coefficient of a road surface. While this approach can achieve short-term (<1 s) motion prediction, it is incapable of predicting motion changes due to certain maneuvers (e.g., sudden slowing down) or interactions with neighboring vehicles (e.g., braking for the leading vehicle).

Maneuver-based methods can compensate for physics-based methods by using drivers' maneuvers (e.g., go straight or turn left or right) in predicting vehicle trajectories. To list some examples, Mandalia et al. [15] use support vector machines to infer driver intentions with a focus on lane-changing decisions. Schreier et al. [16] propose a Bayesian method to predict long-term vehicle trajectories and provide a criticality assessment of the prediction results. Tomar et al. [17] adopt multilayer perceptrons to forecast vehicle trajectories during lane changing.

Most physics-based and maneuver-based approaches do not account for interactions among vehicles. This has motivated the development of interaction-aware methods that take into account the interdependencies of vehicle maneuvers for trajectory prediction. To provide a few examples, Gindele et al. [18] model the mutual influence between vehicles using factored states in prediction. Lefèvre et al. [19] study the joint motion and conflicting intentions of vehicles while assessing the operation risk of a vehicle at the intersection.

Vehicle Trajectory Prediction Using Deep Learning

A number of studies have applied deep learning—especially RNN and its variant, LSTM—for vehicle trajectory prediction [8]–[12], [20]. For example, Deo and Trivedi [11] use a convolutional social pooling network combined with LSTMs to predict vehicle trajectories on highways. Altché and de La Fortelle [12] apply LSTMs to predicting the longitudinal velocity of a vehicle on a highway segment by taking the trajectories of its nine surrounding vehicles into account. Lee et al. [21] propose a deep stochastic inverse optimal control RNN encoder–decoder framework to predict the trajectories of interacting road users in dynamic scenes. Kim et al. [22] propose an LSTM-based trajectory prediction approach using an occupancy grid map to characterize a driving environment.

Attention Mechanisms

Attention mechanisms proposed by Bahdanau et al. [23] can be naturally integrated with RNN to improve the model explainability. For example, Zhou et al. [24] propose an attention-based bidirectional LSTM model to capture key semantic information for relation classification in natural language processing. Lin et al. [25] apply an LSTM model with attention mechanisms to address time series for explainable disease classification.

Attention mechanisms have been used in pedestrian trajectory prediction. Fernando et al. [26] equip LSTMs with soft and hard attention mechanisms to predict the pedestrian trajectory. The soft attention mechanism focuses on the target pedestrian while the hard attention mechanism focuses on neighboring pedestrians. Nevertheless, this method does not capture the interactions between the target pedestrian and neighboring pedestrians. Zhang et al. [27] also propose an attention mechanism for pedestrian trajectory prediction. Their method enables the interpretation of the neighboring pedestrians' effect on the target pedestrian at the spatial level. To the best of our knowledge, our technique is among the few that apply LSTMs with spatiotemporal attention mechanisms at both the spatial level and the temporal level for vehicle trajectory prediction.

Methodology

Following the same setting proposed by Deo and Trivedi [11], we first discretize the space centered around the target vehicle into a 3 × 13 grid. The rows represent the left, current, and right lanes with respect to the target vehicle's location. The columns represent the discretized grid cells with a width of 4.6 m (15 feet) each.

Vehicles that are located inside the 3 × 13 grid (except the target vehicle) are considered neighboring vehicles. Each neighboring vehicle is assigned to a unique grid cell using its front bumper position. For example, a neighboring vehicle located 11 m in front of the target vehicle will be assigned to the third cell ($3 = \lceil 11/4.6 \rceil$) ahead of the target vehicle's cell.

The inputs to our STA-LSTM model are the T-step historical trajectories of all of the vehicles within the 3 × 13 grid. Each vehicle's trajectory is processed by its corresponding LSTM model. The output is an H-step predicted trajectory of the target vehicle. During this process, both temporal-level and spatial-level attention weights are learned. The temporal-level attention weights can be used to analyze the influence of historical trajectories from both the target and neighboring vehicles on prediction. The spatial-level attention weights can be used to explain the influence of neighboring vehicles on prediction. Next, we introduce how these attention weights are computed.

Temporal-Level Attention Calculation

At time step t, the T-step historical trajectory $\{X_{t-T+1}^{v}, \ldots, X_{t}^{v}\}$ of vehicle v (v can be either the target vehicle or a neighboring vehicle) is taken as the input to an LSTM model. Consider the hidden states of the LSTM model $S_t^v = \{h_{t-T+1}^v, \ldots, h_j^v, \ldots, h_t^v\}, S_t^v \in \mathbb{R}^{d \times T}, h_j^v \in \mathbb{R}^{d \times 1}$, where d is the hidden state length. After these hidden states are generated, the temporal attention weights associated with v, $A_t^v = \{\alpha_{t-T+1}^v, \ldots, \alpha_j^v, \ldots, \alpha_t^v\}$, are computed as follows:

$$A_t^v = \mathrm{softmax}(\tanh(W_\alpha S_t^v)), A_t^v \in \mathbb{R}^{1 \times T}, W_\alpha \in \mathbb{R}^{1 \times d}, \quad (1)$$

where W_α represents learnable weights.

Next, we combine the hidden states S_t^v and temporal attention weights A_t^v to derive the tensor-cell value associated with v:

$$H_t^v = S_t^v (A_t^v)^T = \sum_{j=t-T+1}^{t} \alpha_j^v h_j^v, H_t^v \in \mathbb{R}^{d \times 1}. \quad (2)$$

Collectively, all tensor-cell values are adopted to compute the spatial-level attention weights and predict the target vehicle's trajectory.

Spatial-Level Attention Calculation

We can represent all tensor-cell values at t as $G_t = \{G_t^1, \ldots, G_t^n, \ldots, G_t^N\}, G_t \in \mathbb{R}^{d \times N}, G_t^n \in \mathbb{R}^{d \times 1}$, where N represents the total number of tensor cells (i.e., 39). G_t^n takes the following form:

$$G_t^n = \begin{cases} H_t^v, \text{if any vehicle } v \text{ locates at grid cell } n, \\ 0 \in \mathbb{R}^{d \times 1}, \text{ otherwise.} \end{cases} \quad (3)$$

Then, the spatial-level attention weights associated with all vehicles at t, $B_t = \{\beta_t^1, \ldots, \beta_t^n, \ldots, \beta_t^N\}, B_t \in \mathbb{R}^{1 \times N}$, are calculated as follows:

$$B_t = \mathrm{softmax}(\tanh(W_\beta G_t)), W_\beta \in \mathbb{R}^{1 \times d}, \quad (4)$$

where W_β represents learnable weights. Finally, we combine all of the historical information from the target and neighboring vehicles as follows:

$$V_t = G_t (B_t)^T = \sum_{n=1}^{N} \beta_t^n G_t^n. \quad (5)$$

V_t is then fed into a feedforward network to predict the H-step trajectory of the target vehicle $\{\hat{X}_{t+1}^{\text{target}}, \ldots, \hat{X}_{t+H}^{\text{target}}\}$. The whole process along with the architecture of our STA-LSTM model is illustrated in Figure 1.

Experiments

Data Introduction and Model Setup

STA-LSTM is learned and evaluated using the NGSIM data set [13]. The data set consists of vehicle trajectories from the segments of U.S. Highway 101 (US-101) and Interstate 80 (I-80) in the United States. The US-101 segment has a length of 482 m (0.3 mi) and five lanes. The I-80 segment has a length of 644 m (0.4 mi) and six lanes. The data from either US-101 or I-80 contain vehicle trajectories sampled at 10 Hz for 45 min. Each 45-min data set consists of three 15-min subsets recorded over different time spans. This gives us in total six 15-min trajectory subsets for learning and testing STA-LSTM. Since these trajectory data are collected on highways, they contain only forward-moving and lane-changing behaviors. We split each of the six 15-min trajectory subsets into training, validation, and test data sets as 0.7:0.1:0.2. As a result, the training, validation, and test data set have 5,922,867, 859,769, and 1,505,756 entries, respectively. No extra preprocessing (e.g., normalization) is applied to the data set.

To compare our model with the state-of-the-art convolutional social (CS)-LSTM model by Deo and Trivedi [11], we follow the same data processing procedures as theirs. Specifically, we first downsample each vehicle trajectory by a factor of 2. Second, based on the vehicle coordinate (x, y), where the y-axis represents the motion direction of the highway, we discretize the space centered around the target vehicle as a 3 × 13 grid.

We choose the time step to be 0.2 s. Fifteen-step (i.e., 3 s, $T = 15$) historical trajectories of the target and its neighboring vehicles (denoted by v) within the 3 × 13 grid, e.g., $\{X_{t-T+1}^v, \ldots, X_j^v, \ldots, X_t^v\}$ in Figure 1 ($X_j^v = [x_j^v, y_j^v]$), are taken as the inputs to STA-LSTM for predicting the five-step (i.e., 1 s, $H = 5$) future trajectory of the target vehicle.

The goal of STA-LSTM is to minimize the following cost function:

$$\min \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \sum_{j=1}^{H} (\hat{X}_j^i - X_j^i)^2, \quad (6)$$

where N_{train} denotes the training set; $\hat{X}_j^i = [\hat{x}_j^i, \hat{y}_j^i]$ is the predicted position at the jth time step; and $X_j^i = [x_j^i, y_j^i]$ is the actual position at the jth time step.

The hyperparameters of STA-LSTM are optimized using a grid search. The dimension of the embedding space is set to 32. The dimension of the hidden vector of the LSTM model is set to 64. The feedforward layer contains one hidden layer with a dimension of 128. The optimization method Adam [28] is chosen with the learning rate of 0.001. The number of training epochs is set to 10. All experiments are conducted using an Intel(R) Xeon(TM) W-2123 CPU, an Nvidia GTX 1080 GPU, and 32 G RAM. The total training time of STA-LSTM is around 5 h.

STA-LSTM not only achieves comparable prediction accuracy to other state-of-the-art techniques but also explains the influence of historical trajectories and neighboring vehicles on the target vehicle via attention weights.

Prediction Accuracy Comparison

To evaluate STA-LSTM, we implement four benchmark models. The first is CS-LSTM [11], which offers state-of-the-art performance on vehicle trajectory prediction. The second is an LSTM model trained solely using the target vehicle's historical trajectories. We refer to this model as naïve LSTM, the goal of which is to test whether the historical trajectories of neighboring vehicles can be used to improve the prediction accuracy. The third is an LSTM model with only the spatial-level attention mechanism [29]: the last hidden state, which contains the most recent trajectory information, is selected

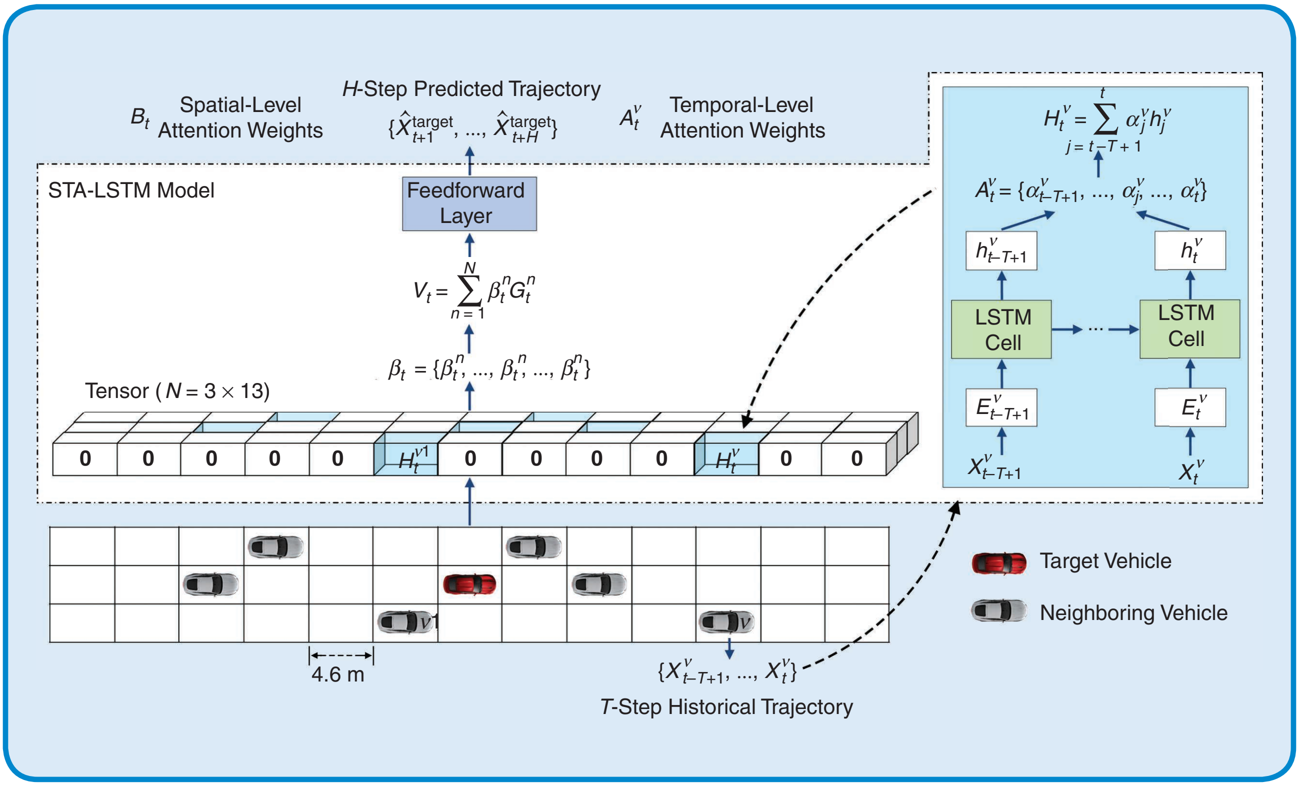

B_t Spatial-Level Attention Weights
H-Step Predicted Trajectory
$\{\hat{X}_{t+1}^{\text{target}}, \ldots, \hat{X}_{t+H}^{\text{target}}\}$
A_t^v Temporal-Level Attention Weights
STA-LSTM Model
Feedforward Layer
$V_t = \sum_{n=1}^{N} \beta_t^n G_t^n$
$\beta_t = \{\beta_t^n, \ldots, \beta_t^n, \ldots, \beta_t^n\}$
Tensor ($N = 3 \times 13$)
0
H_t^{v1}
H_t^v
$H_t^v = \sum_{j=t-T+1}^{t} \alpha_j^v h_j^v$
$A_t^v = \{\alpha_{t-T+1}^v, \ldots, \alpha_j^v, \ldots, \alpha_t^v\}$
h_{t-T+1}^v
h_t^v
LSTM Cell
E_{t-T+1}^v
E_t^v
X_{t-T+1}^v
X_t^v
v1
v
4.6 m
$\{X_{t-T+1}^v, \ldots, X_t^v\}$
T-Step Historical Trajectory
Target Vehicle
Neighboring Vehicle

FIG 1 The schematic view of our approach and the architecture of STA-LSTM. The input to STA-LSTM are the T-step historical trajectories of all of the vehicles within the 3 × 13 grid centered around the target vehicle. Each trajectory is processed by an LSTM model. An example at time step t involving vehicle v is shown (v can be either the target vehicle or a neighboring vehicle). The trajectory $\{X_{t-T+1}^v, \ldots, X_t^v\}$ is used to generate the hidden states $\{h_{t-T+1}^v, \ldots, h_t^v\}$, which are then used to compute the temporal-level attention weights associated with each vehicle, denoted by A_t^v. Next, $\{h_{t-T+1}^v, \ldots, h_t^v\}$ are combined with A_t^v to derive a cell value of the 3 × 13 tensor denoted by H_t^v. After filling the tensor with either H_t^v (has a vehicle) or **0** (no vehicle), we compute the spatial-level attention weights associated with all vehicles B_t and predict the H-step trajectory of the target vehicle $\{\hat{X}_{t+1}^{\text{target}}, \ldots, \hat{X}_{t+H}^{\text{target}}\}$. Note that each vehicle is assigned to a unique grid cell using the front bumper position.

to form the spatial-level attention layer and fuse information from both the target and neighboring vehicles. We refer to this model as SA-LSTM, the goal of which is to test whether including temporal-level attention (in addition to spatial-level attention) will affect the prediction result. The training time of SA-LSTM and CS-LSTM are similar to STA-LSTM (i.e., 5 h). The training time of naïve LSTM is around 3 h since it does not build LSTMs for neighboring vehicles.

Note that because our data set is missing the kinematic and dynamic constraints of the contained heterogeneous traffic and physics-related environmental factors, such as the road's friction coefficient, it is impractical to implement a complex physics-based model for comparison. Nevertheless, since physics-based models can be used in prediction over a short time horizon, for completeness, we build a simple physics-based model to predict the target vehicle's trajectory by extrapolating the historical trajectory under constant longitudinal and lateral speeds.

We measure the performance using the root-mean-square error (RMSE) between the predicted and actual positions of the target vehicle for five time steps at 0.2 s/step. The results are displayed in Table 1. The physics-based model performs the worst among all models. STA-LSTM performs slightly better than CS-LSTM across all time steps. SA-LSTM performs a little worse than STA-LSTM and CS-LSTM. The naïve LSTM, which relies solely on the information of the target vehicle, has the worst performance among the learning-based models. These results indicate that 1) it is helpful to consider the information of neighboring vehicles for vehicle trajectory prediction; 2) it might be sufficient to use the most recent trajectories for prediction; and 3) computing the spatial–temporal attention will not affect the prediction accuracy. Although our STA-LSTM model does not improve the prediction accuracy of CS-LSTM significantly, the learned spatial–temporal attention weights provide interpretability on the prediction results.

Table 1. A Comparison of our model and four benchmark models using RMSE.

	RMSE Per Prediction Time Step (0.2 s)				
Models	1st	2nd	3rd	4th	5th
Physics-based model	0.1776	0.3852	0.6033	0.8377	1.0888
Naïve LSTM	0.1012	0.2093	0.3384	0.4830	0.6406
SA-LSTM	0.1026	0.2031	0.3157	0.4367	0.5643
CS-LSTM [11]	0.1029	0.2023	0.3146	0.4364	0.5674
STA-LSTM (Ours)	0.0995	0.2002	0.3130	0.4348	0.5615

Attention Weights Analysis

Temporal-Level Attention

We start by analyzing the temporal-level attention mechanism. We compute the temporal-level attention weights of 15 historical time steps (from $t-14$ to t) using each of the six 15-min subsets. Figure 2 illustrates the averaged weights from $t-5$ to t. The weights before $t-5$ are omitted as they are negligible. The attention weights at the current time step t are the largest. This indicates that the future trajectory of the target vehicle is mainly influenced by the most recent trajectories of itself and the neighboring vehicles. This result also explains why SA-LSTM, which includes the spatial-level but not the temporal-level attention mechanism, performs only moderately worse than STA-LSTM.

Spatial-Level Attention by Vehicle Class

We next analyze the spatial-level attention mechanism. For convenience, we label each grid cell by its lane name and relative order to the target vehicle's cell. For example, $(Current, 6)$ represents the sixth grid cell in the current lane and ahead of the target vehicle's cell.

Our analysis is based on two main target vehicle types in the NGSIM data set: autos and trucks. The target vehicle's cell has the largest attention weight: 72.14% for autos and 79.53% for trucks. This result, combined with the previous temporal-level attention analysis, reveals that the future trajectory of the target vehicle largely depends on its own driving status. The larger influence of a truck on itself may be because the truck needs a longer time to react to

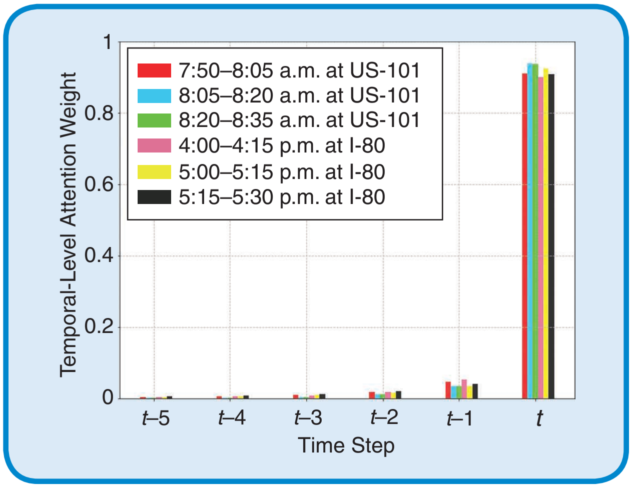
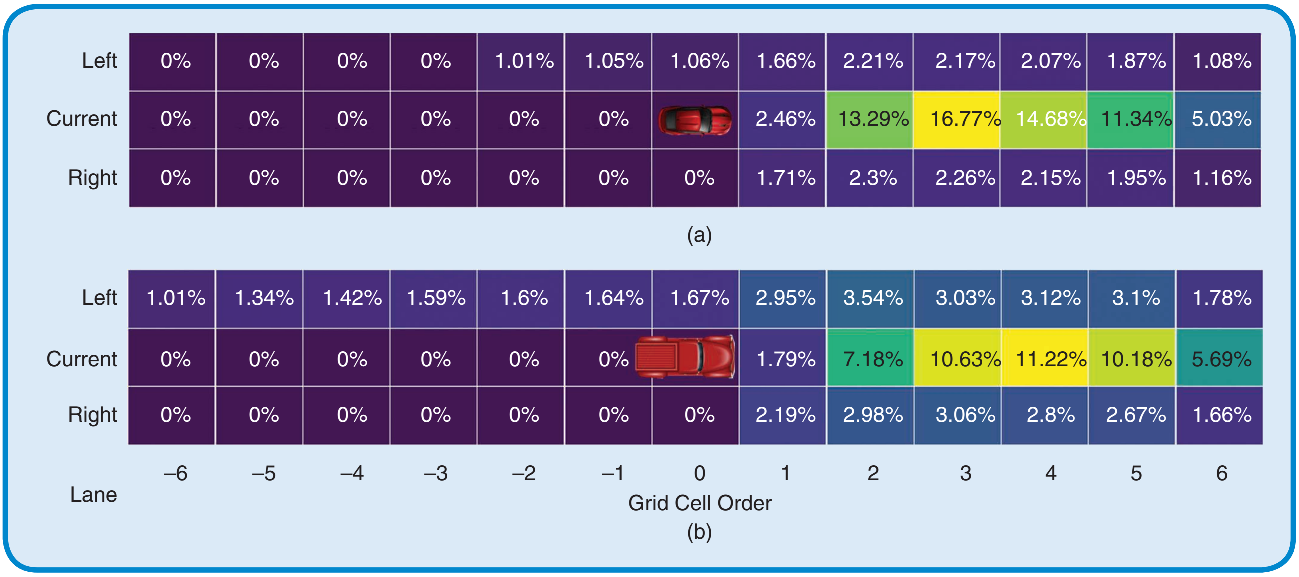


FIG 2 The averaged temporal-level attention weights of six time steps computed using each of the six 15-min subsets. The weights before $t-5$ are omitted as they are negligible. The weights at the current time step t are the largest. This indicates that the future trajectory of the target vehicle is mainly impacted by the most recent trajectories of itself and its neighboring vehicles. In addition, this explains why, by excluding temporal-level attention mechanisms, the performance of SA-LSTM drops only moderately compared to STA-LSTM.



Lane	−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6
Left	0%	0%	0%	0%	1.01%	1.05%	1.06%	1.66%	2.21%	2.17%	2.07%	1.87%	1.08%
Current	0%	0%	0%	0%	0%	0%		2.46%	13.29%	16.77%	14.68%	11.34%	5.03%
Right	0%	0%	0%	0%	0%	0%	0%	1.71%	2.3%	2.26%	2.15%	1.95%	1.16%

(a)

Lane	−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6
Left	1.01%	1.34%	1.42%	1.59%	1.6%	1.64%	1.67%	2.95%	3.54%	3.03%	3.12%	3.1%	1.78%
Current	0%	0%	0%	0%	0%	0%		1.79%	7.18%	10.63%	11.22%	10.18%	5.69%
Right	0%	0%	0%	0%	0%	0%	0%	2.19%	2.98%	3.06%	2.8%	2.67%	1.66%

Grid Cell Order

(b)

FIG 3 The distributions of spatial-level attention weights by target vehicle class (excluding weights in the target vehicle's cell) for (a) autos and (b) trucks. For all cases, cells behind the target vehicle's cell receive virtually no attention weights, showing the negligible influence of vehicles in the back of the target vehicle. For the auto, the largest weights appear at (*Current*, 2), (*Current*, 3), and (*Current*, 4). For the truck, the largest weights appear at (*Current*, 3), (*Current*, 4), and (*Current*, 5), while (*Current*, 1) and (*Current*, 2) receive fewer weights compared to the auto class. This discrepancy may be because the truck often maintains a longer distance from the front vehicle for safety concerns, thus focusing on front vehicles at a further distance. Note that we use vehicles' front bumper positions to compute their belonging cells, and one vehicle contributes only to one cell.

neighboring vehicles. So, its own trajectory plays a heavier role in trajectory prediction. In contrast, an auto is more flexible and can react faster to its neighboring vehicles by altering its trajectory.

To better depict the distribution of attention weights of neighboring vehicles, we normalize and plot the rest of the attention for autos (27.86%) and trucks (20.47%) on the 3 × 13 grid. These results are found in Figure 3(a) and (b). The grid cells behind the target vehicle's cell receive virtually no attention, indicating the negligible influence of following vehicles on the target vehicle. This may be because drivers pay much less attention to following vehicles.

We further observe that when the target vehicle is an auto, all front grid cells on the current lane receive attention weights. The grid cells receiving larger values are (*Current*, 2), (*Current*, 3), and (*Current*, 4). When the target vehicle is a truck, the larger weights are found at (*Current*, 3), (*Current*, 4), and (*Current*, 5), while (*Current*, 1) and (*Current*, 2) receive less weight compared to an auto. This may be because the truck usually keeps a longer distance from the front vehicle to maintain safety and subsequently pays more attention to front vehicles at a further distance. To verify our hypothesis, we calculate the distance from the front bumper of the target vehicle to the back bumper of the front vehicle by vehicle type. The statistics are displayed in Table 2. We can observe that a truck keeps a longer distance to its front vehicle compared to an auto. This explains why (*Current*, 1) and (*Current*, 2) of the truck receive fewer weights compared to the auto, as illustrated in Figure 3.

Spatial-Level Attention by Neighboring Vehicle Density

The NGSIM data set records vehicle trajectories under different traffic conditions. So, it is possible to explore the influence of neighboring vehicle densities on the distribution of spatial-level attention weights.

Because the average number of neighboring vehicles within the 3 × 13 grid is computed as seven, we consider two neighboring vehicle densities: ≤7 and >7. The results are illustrated in Figure 4. When the number is >7, i.e., more congested traffic, the weight of the target vehicle's cell decreases from 75 to 68%, demonstrating the gain of the influence from neighboring vehicles. When the number of neighboring vehicles is ≤7, among the neighboring vehicles, the largest attention weight appears at (*Current*, 4). In contrast, when the number is >7, (*Current*, 2) has the largest attention weight. This may be because when congestion develops, the target vehicle is closer to the front vehicle.

Table 2. The statistics of the distance from the front bumper of the target vehicle to the back bumper of the front vehicle in our data set.

	Front Vehicle Type	
Target Vehicle Type	Auto	Truck
Auto	10.54 ± 5.32 (m)	8.14 ± 4.64 (m)
Truck	12.70 ± 5.42 (m)	14.57 ± 3.80 (m)

These results demonstrate that STA-LSTM can be used to capture driving attentions, including staying in the same lane and switching lanes.

Spatial-Level Attention by Location

In the NGSIM data set, the study segment of US-101 consists of five lanes, and the segment of I-80 contains six lanes. Each segment contains one additional ramp lane. These configurations allow us to analyze the distribution of the maximum spatial-level attention weight (of neighboring vehicles), especially when the target vehicle is in different lanes.

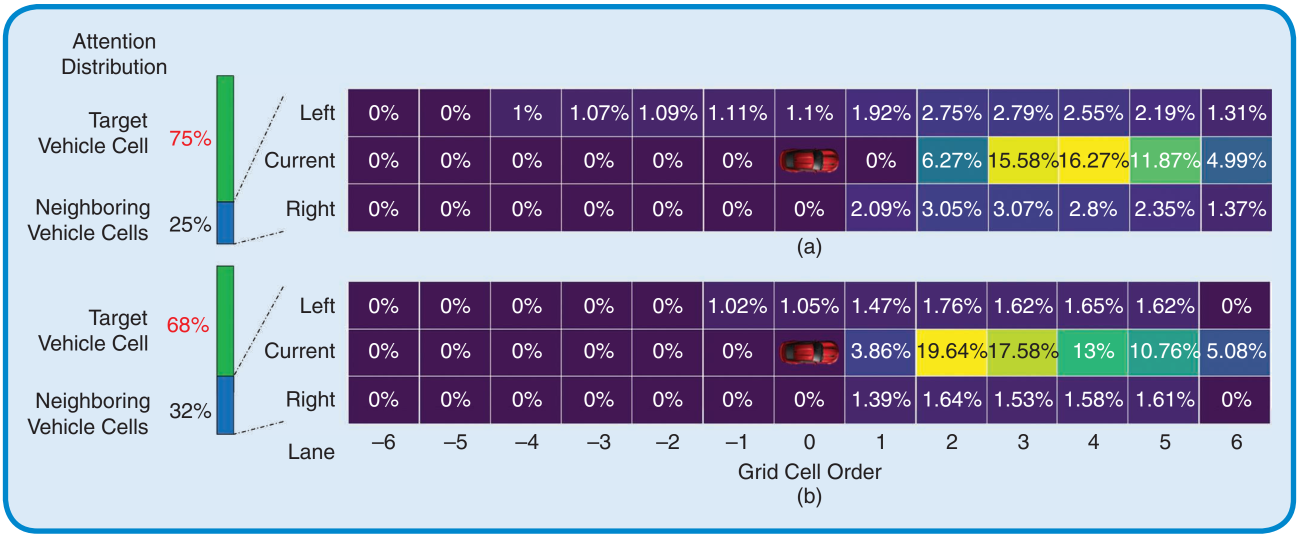
Here, we use the case of US-101 as an example. We select four lanes from US-101 southbound: the innermost lane, middle lane, outermost lane, and ramp segment. These lanes are shown in Figure 5(a). Figure 5(b) portrays grid cells with the frequency counts—each count indicates that one maximum spatial-level attention weight regarding a neighboring vehicle was assigned to this cell.

As we can see from Figure 5(b), target vehicles mainly focus on front vehicles in the current lane. An exception is the ramp segment, where target vehicles pay more attention to front vehicles in the left lane, indicating their intention to switch to it. Target vehicles in the outermost lane also pay more attention to the lane on the left compared to the lane on the right, showing the preference to change to the left rather than right. In contrast, a reverse pattern is found on the innermost lane, where target vehicles pay more attention to front vehicles in the right lane in addition to the current lane. Target vehicles in the middle lane indicate smaller differences in attention distribution between the left and right lanes. These results demonstrate that STA-LSTM can be used to capture driving attentions, including staying in the same lane and switching lanes. Next, we show that STA-LSTM can identify the moment when specific lane-changing behaviors take place.

Spatial-Level Attention on Lane-Changing Behaviors

To study whether spatial-level attention weights can explain specific driving behaviors such as lane changing, we have selected the vehicle with ID 2858 as the study subject, which conducted two lane-changing maneuvers on I-80.

The target vehicle 2858 executes the first lane-changing maneuver around the 996th time step from lane 4 to 5 and the second lane-changing maneuver around the 1,220th time step from lane 5 to 6. This is illustrated in Figure 6. In addition, we show the grid cells containing neighboring vehicles that receive the largest attention weight at each time step during this process in Figure 6. We observe that the target vehicle (i.e., vehicle 2858) mainly focuses on front vehicles in the current lane for the first 977 time steps. From the 978th to the 996th time step, it gradually relocates the maximum attention from the current lane to (*Right*, 1) and then (*Right*, 2) as it prepares to change to the right lane. The maximum attention weight



Lane	–6	–5	–4	–3	–2	–1	0	1	2	3	4	5	6
Left	0%	0%	1%	1.07%	1.09%	1.11%	1.1%	1.92%	2.75%	2.79%	2.55%	2.19%	1.31%
Current	0%	0%	0%	0%	0%	0%		0%	6.27%	15.58%	16.27%	11.87%	4.99%
Right	0%	0%	0%	0%	0%	0%	0%	2.09%	3.05%	3.07%	2.8%	2.35%	1.37%

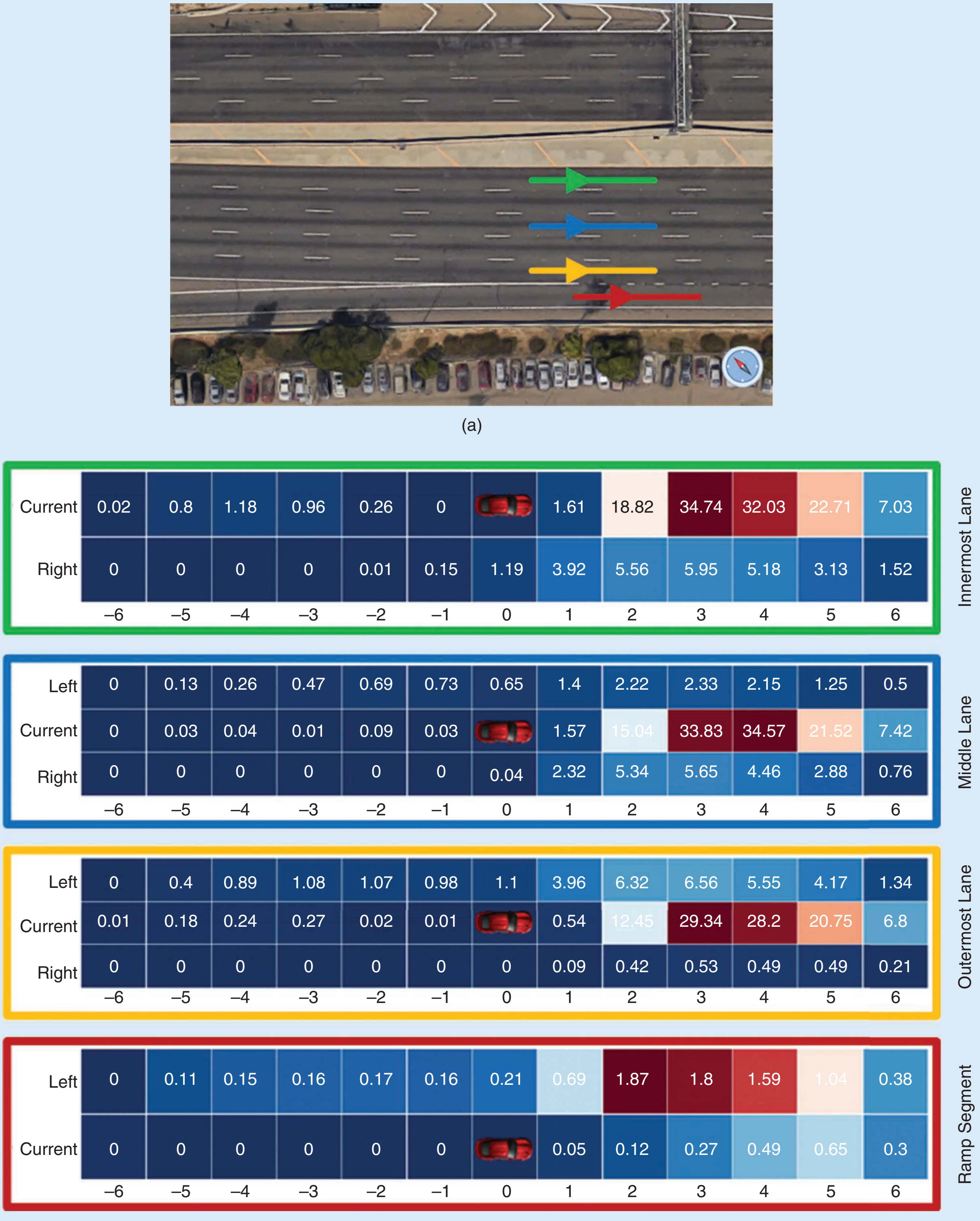
(a)

Lane	–6	–5	–4	–3	–2	–1	0	1	2	3	4	5	6
Left	0%	0%	0%	0%	0%	1.02%	1.05%	1.47%	1.76%	1.62%	1.65%	1.62%	0%
Current	0%	0%	0%	0%	0%	0%		3.86%	19.64%	17.58%	13%	10.76%	5.08%
Right	0%	0%	0%	0%	0%	0%	0%	1.39%	1.64%	1.53%	1.58%	1.61%	0%

Grid Cell Order

(b)

FIG 4 The averaged spatial-level attention weights by vehicle density. (a) Less congested traffic, with the number of neighboring vehicles ≤7 and (b) more congested traffic, with the number of neighboring vehicles >7. As congestion develops, the attention weight of the target vehicle's cell decreases from 75 to 68%, indicating that the neighboring vehicles have more influence on the target vehicle in a congested environment. By normalizing and plotting the attention weights of neighboring vehicles, we can see that the largest attention weight is located at (*Current*, 4) in (a) and at (*Current*, 2) in (b), which may be caused by the shortened distance between the target vehicle and its leading vehicle.



(a)

	−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6
Current	0.02	0.8	1.18	0.96	0.26	0		1.61	18.82	34.74	32.03	22.71	7.03
Right	0	0	0	0	0.01	0.15	1.19	3.92	5.56	5.95	5.18	3.13	1.52

	−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6
Left	0	0.13	0.26	0.47	0.69	0.73	0.65	1.4	2.22	2.33	2.15	1.25	0.5
Current	0	0.03	0.04	0.01	0.09	0.03		1.57	15.04	33.83	34.57	21.52	7.42
Right	0	0	0	0	0	0	0.04	2.32	5.34	5.65	4.46	2.88	0.76

	−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6
Left	0	0.4	0.89	1.08	1.07	0.98	1.1	3.96	6.32	6.56	5.55	4.17	1.34
Current	0.01	0.18	0.24	0.27	0.02	0.01		0.54	12.45	29.34	28.2	20.75	6.8
Right	0	0	0	0	0	0	0	0.09	0.42	0.53	0.49	0.49	0.21

	−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6
Left	0	0.11	0.15	0.16	0.17	0.16	0.21	0.69	1.87	1.8	1.59	1.04	0.38
Current	0	0	0	0	0	0		0.05	0.12	0.27	0.49	0.65	0.3

(b)

FIG 5 The maximum spatial-level attention weight frequency by target vehicle location. (a) Four lanes are selected from US-101 southbound. (Source: Google Maps.) (b) Grid cells are filled with frequency counts (× 1,000) that indicate the locations of the maximum spatial-level attention weights. Except for the ramp segment, target vehicles mainly focus on the current lane. On the ramp segment, the target vehicles pay more attention to the left lane, illustrating their intention to switch to it. The frequency distribution on the other lanes can be interpreted in a similar manner. These results demonstrate that STA-LSTM can capture various driving intentions, such as staying in the same lane and switching lanes.

Our experiment results indicate that STA-LSTM not only achieves performance comparable to other state-of-the-art techniques in prediction accuracy but, more importantly, provides spatial-temporal attention weights for enhancing model explainability.

shifts from the current lane to the right lane again at the 1,183th time step and stays mostly at (*Right*, 1) and (*Right*, 2) until the 1,220th time step. The duration of the maximum attention weight shift indicates that the first lane changing takes 3.8 s (19×0.2), and the second lane changing takes 7.6 s (38×0.2).

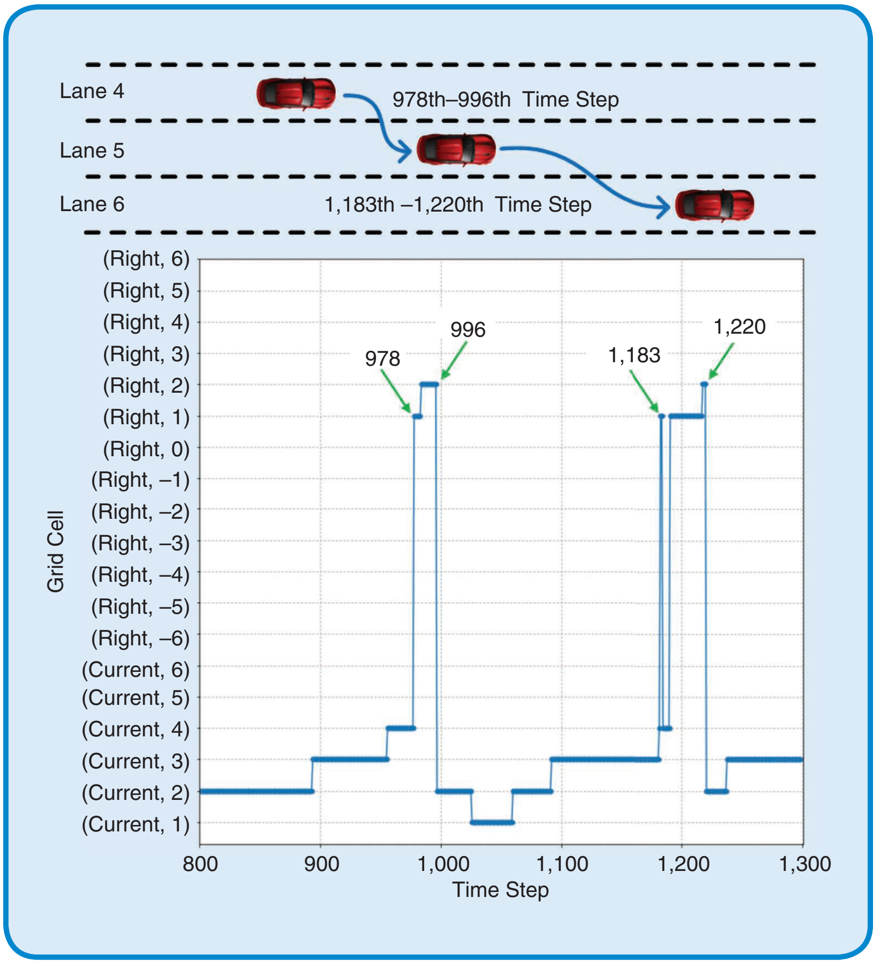


FIG 6 The maximum spatial-level attention weights regarding the lane-changing behaviors of the target vehicle 2858. Two lane-changing maneuvers are executed: one at the 996th time step and the other at the 1,220th time step. Prior to the first lane changing, the target vehicle mainly focuses on the front vehicles in the current lane. During the two lane-changing executions to the right lane, the maximum attention of the target vehicle switches to (*Right*, 1) and (*Right*, 2) (i.e., the two spikes in the diagram). These results demonstrate that STA-LSTM can capture intricate driving behaviors in detail.

To verify whether STA-LSTM can capture lane-changing behaviors correctly with attention weights, the trajectories of vehicle 2858 and its neighboring vehicles are examined. There is a neighboring vehicle with ID 2846 at (*Right*, 0) at the 964th time step, and no other neighboring vehicles are behind vehicle 2846 at the time. Vehicle 2858 first slows down from the 964th to the 967th time step, when the relative position of vehicle 2846 changes from (*Right*, 0) to (*Right*, 1). The speed of vehicle 2858 keeps decreasing from the 968th to the 983th time step until the relative position of the neighboring vehicle 2846 changes to (*Right*, 2). The target vehicle 2858 then starts increasing its speed while maintaining the relative position of vehicle 2846 at (*Right*, 2) before finally changing to the right lane at the 996th time step. A similar pattern is observed during the second lane-changing maneuver. These results demonstrate that STA-LSTM is capable of capturing complex lane-changing maneuvers in detail.

Conclusion

Vehicle trajectory prediction is an essential task for many intelligent transportation system (ITS) applications. The importance of this task is emphasized with the emergence of AVS as they require an interpretable prediction of the future motions of surrounding vehicles to navigate safely and efficiently. We propose STA-LSTM by integrating LSTMs with spatial–temporal attention mechanisms for explainability in vehicle trajectory prediction.

STA-LSTM is learned and evaluated using the NGSIM data set [13], which contains real-world vehicle trajectories from the segments of US-101 and I-80 in the United States. Our experiment results indicate that STA-LSTM not only achieves performance comparable to other state-of-the-art techniques in prediction accuracy but, more importantly, provides spatial–temporal attention weights for enhancing model explainability. The learned attention weights can be used to explain the influences of historical trajectories and the locations of neighboring vehicles on the target vehicle's future motion.

We conduct detailed analyses of the learned attention weights based on various vehicle and environmental factors, including target vehicle class, target vehicle locations, and neighboring vehicle densities. In addition, we find that the learned attention weights can be used to interpret lane-changing behaviors of the target vehicle. Together, our in-depth study of the attention distribution of the target vehicle on itself and neighboring vehicles can potentially benefit the development of many ITS applications, such as advanced driver assistance and AV motion planning and navigation.

> We propose STA-LSTM by integrating LSTMs with spatial–temporal attention mechanisms for explainability in vehicle trajectory prediction.

Future Work

Many future research directions can stem from this work. First, we will continue to test the performance of STA-LSTM over other prediction horizons and analyze the change of the attention weights. Second, instead of using grid-based discretization to model the relationship between the target vehicle and neighboring vehicles, other types of data structures can be explored. For example, a graph in which nodes representing vehicles and edges representing the influences among vehicles can be used to replace the grid. Therefore, it would be interesting to test whether a graph-based deep learning technique such as the graph convolutional neural network [30], [31] can be used to capture the correlations among vehicles and predict vehicle trajectories.

The data used to learn STA-LSTM are from stationary sensors installed on US-101 and I-80. While these sensors provide complete and accurate traffic measurements, they are mostly found on highways and major roads, which constitute only a small portion of a city. To use our approach for autonomous driving on arterial roads, we need to work with mobile data such as GPS reports. Given that GPS data can be either sparsely or densely sampled, it would be interesting to combine the previous techniques for addressing sparse [32] or dense GPS data [33], [34] with STA-LSTM. Finally, it would be interesting to extend our approach to other trajectory data sets through transfer learning [35].

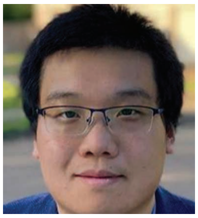
STA-LSTM can also be used to enhance traffic simulation models. Realistic virtual traffic, as a result of an improved simulation technique, has many applications in fields including: 1) ITS, such as analyzing congestion causes, identifying network bottlenecks, and testing transport policies at the macroscopic scale [36]–[38]; and 2) virtual environments, such as improving the believability of traffic animation and reconstruction [39], [40] and enhancing the training and testing of AVs at the microscopic scale [41]. It would be of great use to develop simulation models that incorporate STA-LSTM.

Acknowledgement

We would like to thank the University of Memphis for providing the start-up funds.

About the Authors

Lei Lin (lei.lin@rochester.edu) earned his Ph.D. degree in transportation systems engineering from the University at Buffalo, the State University of New York, Buffalo, in 2015. He is a research scientist at Goergen Institute for Data Science, University of Rochester, Rochester, New York, 14627, USA. His research interests include transportation big data, artificial intelligence applications in transportation, and connected and automated transportation. He is a Member of IEEE.



Weizi Li (wli@memphis.edu) earned his Ph.D. degree in computer science from the University of North Carolina at Chapel Hill. Currently, he is an assistant professor in the Department of Computer Science at the University of Memphis, Memphis, Tennessee, 38152, USA. Prior to this position, he was a Michael Hammer Postdoctoral Fellow at the Institute for Data, Systems, and Society of the Massachusetts Institute of Technology. His current research interests include intelligent transportation systems, multiagent simulation, virtual environments, machine learning, and robotics.

Huikun Bi (bihuikun@ict.ac.cn) earned her Ph.D. degree from the University of Chinese Academy of Sciences. She is an assistant professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China. Her main research interests include crowd simulation, motion forecasting, and deep learning. She is a Member of IEEE.

Lingqiao Qin (Lingqiao.qin@wisc.edu) earned her M.S. degree in transportation safety engineering from George Washington University, Washington, D.C., and her M.S. degree in industrial and systems engineering from University of Wisconsin–Madison. She is currently working toward her Ph.D. degree in transportation engineering at the University of Wisconsin–Madison, Madison, Wisconsin, 53704, USA. Her research interests include traffic operations, the next generation of transportation (autonomous and connected vehicles), and using advanced technologies such as driving simulators and eye trackers to improve the design, operations, and safety of all elements in transportation. She is a Member of IEEE.

References

[1] G. Seetharaman, A. Lakhotia, and E. P. Blasch, “Unmanned vehicles come of age: The DARPA grand challenge,” *Computer*, vol. 39, no. 12, pp. 26–29, 2006. doi: 10.1109/MC.2006.447.

[2] D. Pomerleau, “ALVINN: An autonomous land vehicle in a neural network,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1989, pp. 305–313.

[3] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, “Off-road obstacle avoidance through end-to-end learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 739–746.

[4] M. Bojarski et al., “End to end learning for self-driving cars,” 2016, arXiv:1604.07316.

[5] W. Li, D. Wolinski, and M. C. Lin, “ADAPS: Autonomous driving via principled simulations,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2019, pp. 7625–7631.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013. doi: 10.1177/0278364913491297.

[7] Y. Shen, W. Li, and M. C. Lin, “Autonomous driving via multi-sensor perception and weighted inverse reinforcement learning,” Univ. of Maryland, College Park, Tech. Rep., 2020.

[8] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F.-Y. Wang, “Capturing car-following behaviors by deep learning,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 910–920, 2017. doi: 10.1109/TITS.2017.2706963.

[9] M. Zhou, X. Qu, and X. Li, “A recurrent neural network based microscopic car following model to predict traffic oscillation,” *Transp. Res. C, Emerg. Technol.*, vol. 84, pp. 245–264, Nov. 2017. doi: 10.1016/j.trc.2017.08.027.

[10] X. Huang, J. Sun, and J. Sun, “A car-following model considering asymmetric driving behavior based on long short-term memory neural networks,” *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 346–362, Oct. 2018. doi: 10.1016/j.trc.2018.07.022.

[11] N. Deo and M. M. Trivedi, “Convolutional social pooling for vehicle trajectory prediction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1468–1476. doi: 10.1109/CVPRW.2018.00196.

[12] F. Altché and A. de La Fortelle, “An LSTM network for highway trajectory prediction,” in *Proc. 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 353–359. doi: 10.1109/ITSC.2017.8317913.

[13] Next generation simulation (NGSIM). Federal Highway Administration. https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

[14] S. Lefèvre, D. Vasquez, and C. Laugier, “A survey on motion prediction and risk assessment for intelligent vehicles,” *ROBOMECH J.*, vol. 1, no. 1, 2014. doi: 10.1186/s40648-014-0001-z.

[15] H. M. Mandalia and M. D. D. Salvucci, “Using support vector machines for lane-change detection,” *Proc. Human Factors Ergonom. Soc. Annu. Meeting*, vol. 49, no. 22, pp. 1965–1969, 2005. doi: 10.1177/154193120504902217.

[16] 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.*, 2014, pp. 334–341. doi: 10.1109/ITSC.2014.6957713.

[17] R. S. Tomar and S. Verma, “Safety of lane change maneuver through a priori prediction of trajectory using neural networks,” *Netw. Protocols Algorithms*, vol. 4, no. 1, pp. 4–21, 2012. doi: 10.5296/npa.v4i1.1240.

[18] T. Gindele, S. Brechtel, and R. Dillmann, “A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments,” in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, 2010, pp. 1625–1631. doi: 10.1109/ITSC.2010.5625262.

[19] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmàn, “Intention-aware risk estimation for general traffic situations, and application to intersection safety,” Ph.D. dissertation, INRIA, 2013.

[20] L. Lin, S. Gong, T. Li, and S. Peeta, “Deep learning-based human-driven vehicle trajectory prediction and its application for platoon control of connected and autonomous vehicles,” in *Proc. Autonom. Veh. Symp.*, vol. 2018, 2018.

[21] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, “Desire: Distant future prediction in dynamic scenes with interacting agents,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 336–345. doi: 10.1109/CVPR.2017.233.

[22] B. Kim et al., “Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network,” 2017, arXiv:1704.07049.

[23] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015.

[24] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, “Attention-based bidirectional long short-term memory networks for relation classification,” in *Proc. 54th Annu. Meeting Assoc. Computat. Linguist. (Volume 2: Short Papers)*, 2016, vol. 2, pp. 207–212. doi: 10.18653/v1/P16-2034.

[25] L. Lin, B. Xu, W. Wu, T. Richardson, and E. A. Bernal, “Medical time series classification with hierarchical attention-based temporal convolutional networks: A case study of myotonic dystrophy diagnosis,” 2019, arXiv:1903.11748.

[26] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, “Soft+hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection,” *Neural Netw.*, vol. 108, pp. 466–478, 2018. doi: 10.1016/j.neunet.2018.09.002.

[27] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, “SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 12,085–12,094. doi: 10.1109/CVPR.2019.01236.

[28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, 2015.

[29] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” in *Proc. 2016 Conf. North Amer. Chap. Assoc. Computat. Linguist., Human Language Technol.*, 2016, pp. 1480–1489. doi: 10.18653/v1/N16-1174.

[30] L. Lin, Z. He, and S. Peeta, “Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach,” *Transp. Res. C, Emerg. Technol.*, vol. 97, pp. 258–276, Dec. 2018. doi: 10.1016/j.trc.2018.10.011.

[31] L. Lin, W. Li, and S. Peeta, “Predicting station-level bike-sharing demands using graph convolutional neural network,” in *Proc. Transp. Res. Board 98th Annu. Meeting (TRB)*, 2019.

[32] W. Li, D. Nie, D. Wilkie, and M. C. Lin, “Citywide estimation of traffic dynamics via sparse GPS traces,” *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 100–113, 2017. doi: 10.1109/MITS.2017.2709804.

[33] L. Lin, W. Li, and S. Peeta, “Efficient data collection and accurate travel time estimation in a connected vehicle environment via real-time compressive sensing,” *J. Big Data Anal. Transp.*, vol. 1, nos. 2–3, pp. 95–107, 2019. doi: 10.1007/s42421-019-00009-5.

[34] L. Lin, S. Peeta, and J. Wang, “Efficient collection of connected vehicle data based on compressive sensing,” in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 3427–3432. doi: 10.1109/ITSC.2018.8570007.

[35] H.-C. Shin et al., “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning,” *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, 2016. doi: 10.1109/TMI.2016.2528162.

[36] D. Wilkie, J. Sewall, W. Li, and M. C. Lin, “Virtualized traffic at metropolitan scales,” *Front. Robot. AI*, vol. 2, p. 11, May 2015. doi: 10.3389/frobt.2015.00011.

[37] W. Li, D. Wolinski, and M. C. Lin, “City-scale traffic animation using statistical learning and metamodel-based optimization,” *ACM Trans. Graph*, vol. 36, no. 6, pp. 200:1–200:12, Nov. 2017. doi: 10.1145/3130800.3130847.

[38] W. Li, M. Jiang, Y. Chen, and M. C. Lin, “Estimating urban traffic states using iterative refinement and wardrop equilibria,” *IET Intell. Transp. Syst.*, vol. 12, no. 8, pp. 875–883, 2018. doi: 10.1049/iet-its.2018.0007.

[39] H. Bi, T. Mao, Z. Wang, and Z. Deng, “A data-driven model for lane-changing in traffic simulation,” in *Proc. Symp. Comput. Anim.*, 2016, pp. 149–158.

[40] Q. Chao, Z. Deng, J. Ren, Q. Ye, and X. Jin, “Realistic data-driven traffic flow animation using texture synthesis,” *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 2, pp. 1167–1178, 2017. doi: 10.1109/TVCG.2017.2648790.

[41] Q. Chao, H. Bi, W. Li, T. Mao, Z. Wang, M. C. Lin, and Z. Deng, “A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving,” *Comput. Graph. Forum*, vol. 39, no. 1, pp. 287–308, 2019. doi: 10.1111/cgf.13803.

ITS