2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Online Prediction of Lane Change with a Hierarchical Learning-Based Approach

Xishun Liao*, Ziran Wang†, Xuanpeng Zhao*, Zhouqiao Zhao*, Kyungtae Han†, Prashant Tiwari†, Matthew J. Barth*, and Guoyuan Wu*

Abstract— **In the foreseeable future, connected and automated vehicles (CAVs) and human-driven vehicles will share the road networks together. In such a mixed traffic environment, CAVs need to understand and predict maneuvers of surrounding vehicles for safer and more efficient interactions, especially when human drivers bring in a wide range of uncertainties. In this paper, we propose a learning-based lane-change prediction algorithm that considers the driving behaviors of the target human driver. To provide accurate maneuver prediction, we adopt a hierarchical structure that seamlessly seals both the lane-change decision prediction and the vehicle trajectory prediction together. Specifically, we propose a lane-change decision prediction method based on a Long-Short Term Memory (LSTM) network, and a trajectories prediction considering driver preference and vehicular interactions based on Inverse Reinforcement Learning (IRL). To validate the performance of the proposed methodology, a case study of an on-ramp merging scenario is conducted on a uniquely built human-in-the-loop simulation platform that can provide an immersive driving environment, collect data of lane-change behaviors, and test drivers' reactions to the prediction results in real time. It is shown in the simulation results that we can predict the lane-change decision 3 seconds before the vehicle crosses the line to another lane, and the Mean Euclidean Distance between the predicted trajectory and ground truth is 0.39 meters within a 4-second prediction window.**

I. Introduction

The emergence of connected and automated vehicles (CAVs) over the last decades brings new solutions to address the safety, mobility, and environmental sustainability issues of our transportation systems [1]. These CAVs can be driven under partial or full automation with the help of their on-board perception sensors, and can also cooperate with other transportation entities through vehicle-to-everything (V2X) communications. Although the CAV technology itself has been evolving rapidly, our transportation systems cannot achieve full automation/connectivity in the very near future. In a mixed traffic environment with CAVs and human-driven vehicles, a CAV actively perceives the surrounding traffic, predicts human-driven vehicles' behaviors, makes decisions regarding the actions to be taken, and executes such actions through the planner and controller. Particularly, the prediction of human-driven vehicles' behaviors is challenging due to the uncertainties of human drivers.

Although lane change is a fundamental maneuver of our daily driving, it is also one of the trickiest since it requires the tacit cooperation of lateral control and longitudinal control from the driver. Therefore, compared with the prediction of longitudinal maneuvers such as car-following, which is heavily correlated with the gap between the ego vehicle and the leading vehicle [2], the prediction of lane change is much more complicated and challenging. The online lane-change prediction of human drivers becomes essential for CAVs, since it provides inputs to the downstream motion planners and controllers and hence allows CAVs to better cooperate with surrounding human-driven vehicles.

Hidden Markov Model (HMM) was widely used to infer the lane-change intention [3] [4] [5] and usually integrated with Bayesian network [6] to recognize the lane-change behavior. As the lane-change intention prediction can be modeled as a classify problem, some researchers [7] [8] used Support Vector Machine (SVM) classifier to distinguish the lane-change maneuver from normal driving state, and similarly, the multilayer perceptron was used as a discriminator [9] in long-term lane-change prediction. Moreover, deep learning methods, such as Long Short-Term Memory (LSTM) model, achieved state-of-art performance on time-series problems, as in [10] lane-change maneuvers can be predicted in 3.5 seconds with a precision of 90.5%.

However, the limitation of labeled data insufficiency is obvious for supervised learning since supervised learning methods require a large enough labeled dataset to cover each possible scenario. To label more data for supervised learning, Mahajan et al. [11] introduced an unsupervised automatic data labeling method and used the customized labeled dataset to train an LSTM network for lane-change prediction. Even if the dataset is well labeled, the data collected by rational and behave drivers cannot explore all corner cases. Thus, Inverse Reinforcement Learning (IRL) [12] becomes popular as it can learn the driver preference, is flexible to unseen circumstances, and can guarantee the feasibility of the predicted trajectories [13]. Furthermore, the prediction should not only consider the vehicle states but also need to consider the vehicular interaction and the driver preference to further investigate lane-change trajectories prediction.

Corresponding author: Xishun Liao, xliao016@ucr.edu

*Department of Electrical and Computer Engineering, and the Center for Environmental Research and Technology, University of California, Riverside, CA 92507.

†InfoTech Labs, Toyota Motor North America R&D, Mountain View, CA 94043.

978-1-7281-9680-0/22/$31.00 ©2022 IEEE

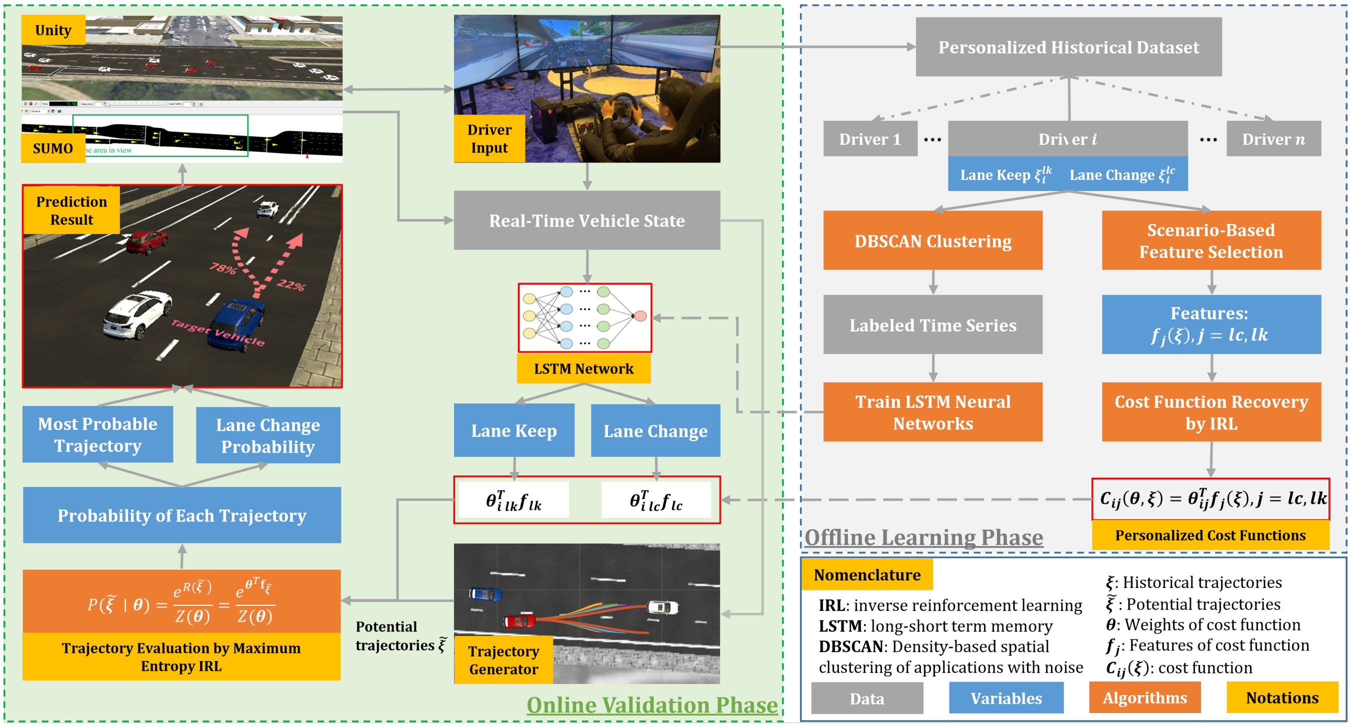

Unity
SUMO
Driver Input
Prediction Result
78%
22%
Target Vehicle
Real-Time Vehicle State
LSTM Network
Lane Keep
Lane Change
$\theta_{i\,lk}^T f_{lk}$
$\theta_{i\,lc}^T f_{lc}$
Most Probable Trajectory
Lane Change Probability
Probability of Each Trajectory
$P(\tilde{\xi} \mid \theta) = \frac{e^{R(\tilde{\xi})}}{Z(\theta)} = \frac{e^{\theta^T f_{\tilde{\xi}}}}{Z(\theta)}$
Trajectory Evaluation by Maximum Entropy IRL
Potential trajectories $\tilde{\xi}$
Trajectory Generator
Online Validation Phase
Personalized Historical Dataset
Driver 1
Driver i
Driver n
Lane Keep ξ_i^{lk}
Lane Change ξ_i^{lc}
DBSCAN Clustering
Scenario-Based Feature Selection
Labeled Time Series
Features: $f_j(\xi), j = lc, lk$
Train LSTM Neural Networks
Cost Function Recovery by IRL
$C_{ij}(\theta, \xi) = \theta_{ij}^T f_j(\xi), j = lc, lk$
Personalized Cost Functions
Offline Learning Phase
Nomenclature
IRL: inverse reinforcement learning
LSTM: long-short term memory
DBSCAN: Density-based spatial clustering of applications with noise
ξ: Historical trajectories
$\tilde{\xi}$: Potential trajectories
θ: Weights of cost function
f_j: Features of cost function
$C_{ij}(\xi)$: cost function
Data
Variables
Algorithms
Notations

Fig. 1. System workflow of the hierarchical learning-based approach for lane-change prediction with an offline learning phase (in grey) and an online validation phase (in green).

In [14], the driver preferences over different behaviors were formulated as a cost function and learned by IRL. To integrate the interactive behaviors into prediction, Huang et al. [15] proposed a sampling-based IRL to learn the reward function for building the highway driving behavior model and predicted the vehicle trajectories by estimating the probability of the sampled trajectories. This method was validated on a highway lane-change prediction case study. Sun et al. [16] introduced a hierarchical IRL-based algorithm to predict both discrete decisions and continuous trajectories of a target vehicle involved in two-vehicle interaction. An offline evaluation was performed on a mandatory lane-change driving scenario.

In this study, we train an LSTM network to provide binary prediction for the lane-change decisions based on historical vehicle states, and recover the driver's preference (i.e., cost functions) based on IRL. Further, the most probable trajectory and the probability of this binary prediction can be evaluated, using the IRL recovered cost functions. The proposed algorithm can be used to model the behavior of human driven vehicles with connectivity (e.g., by cellphone), and the learned driver models are stored on cloud. Every time encountering a connected vehicle in the cloud database, CAVs can identify and obtain its model to facilitate the prediction.

Compared to existing literature on prediction and behavior modeling, the following contributions are made in this study:

- Based on density-based spatial clustering of applications with noise (DBSCAN), we develop a unsupervised data labeling method by adding temporal information to relate adjacent data points in a time series.
- The proposed hierarchical learning-based approach provides predictions for lane-change decisions and vehicle trajectories, along with likelihood estimation.
- The online lane-change prediction is validated in human-in-the-loop simulation using a uniquely built co-simulation platform.

II. Problem Formulation

The proposed system for lane-change behavior prediction is shown in Fig. 1. The system consists of an offline learning process and an online validation process. In the offline learning process, based on the dataset collected on a co-simulation platform, an LSTM model is trained to predict the lane-change decision, and the cost function inferring the driver preference is learned by the IRL. In the online validation process, at each time step, the vehicle states will be analyzed by the LSTM network to recognize the maneuver and select a proper cost function. Next, the cost function evaluates the confidence of possible trajectories provided by the trajectories generator. Finally, the outputs including the most probable trajectory and lane change probability are visualized and sent back to the co-simulation platform, and the major results of each phase are highlighted with red boundary. The offline process containing most of the algorithms is introduced in Section III, while the online validation details are elaborated in Section IV.

This work is an extension of our previous work [17], where a cloud-based Digital Twin framework was designed for the storage of driver data and behavior models. Data fusion technology was implemented to collect historical data via on-board sensors and V2X communications. In this study, the target predicted vehicle is assumed to be a connected human-driven vehicle, whose historical/real-time data and the trained driver model are accessible through the Digital Twin. When other connected vehicles detect this target vehicle, they can download the driver model of the target vehicle to assist the prediction. Specifically, our prediction algorithm is designed for the on/off-ramp scenario to predict maneuver and trajectories of the on-ramp driver, who either performs a mandatory lane change before the end of the merging area, or has to keep his/her current lane and enter the off-ramp.

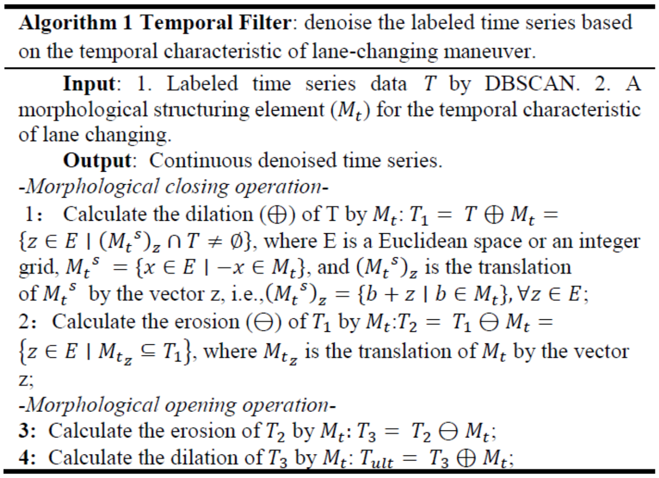
III. Methodology

In this study, we propose an algorithm to predict the lane-change behavior based on the target vehicle's maneuver and its surrounding environment. A behavior model of a driver is built based on the historical trajectories $\Xi=\{\xi_k\}, k=1,\ldots,K$, and a trajectory contains vehicle states at every time step, i.e., $\xi_k=[s_1,s_2,\ldots,s_t]$, where s_{t} is a vector and denotes the vehicle state at t-th time step. The vehicle states consist of the information of ego vehicle and its surrounding environment, which can reflect the operation and the perception of the driver. The future trajectory is denoted as $\hat{\xi}=[s_{t+1},\ldots,s_{t+T}]$, where T is the prediction trajectory horizon. Since the target vehicle's lane-change action and trajectory in future T steps depend on its vehicle states in past steps, we formulate this influence as conditional probability density functions: $p\left(A_{t:t+T}\mid\xi\right)$ and $p\left(\hat{\xi}\mid\xi\right)$, respectively, where $A=\{a_{\text{change}},a_{\text{keep}}\}$, i.e., lane change and lane keep.

A. Unsupervised Data Labeling Based on Modified DBSCAN

To predict the lane-change maneuver, we need to recognize those lane-change moments among the dataset, so labeling each time step for the dataset is the first step of data processing. The raw trajectory data is integrated with the map information to create a dataset that contains features potentially affecting driver's lane-change decision making, and these factors include vehicle speed, position, distance to surrounding vehicles, speed of surrounding vehicles, deviation from the lane centerline, remaining distance to the mandatory lane-change point (if any), and speed limit.

We separate the trajectories that contain lane-change maneuvers from the whole dataset, by monitoring the accumulated lane deviation. Then those trajectories are further processed for recognizing the decision of the driver at each time step. Inspired by [11], we applied DBSCAN to label lane-change and lane-keep maneuvers for each vehicle state at each time step. The lateral speed (v_{lat}) and lateral acceleration (a_{lat}) are used as two input features for DBSCAN, and the outputs are two clusters of vehicle states, e.g., lane change or lane keep.



```
**Algorithm 1 Temporal Filter**: denoise the labeled time series based on the temporal characteristic of lane-changing maneuver.
    **Input**: 1. Labeled time series data $T$ by DBSCAN. 2. A morphological structuring element ($M_t$) for the temporal characteristic of lane changing.
    **Output**: Continuous denoised time series.
-*Morphological closing operation*-
1: Calculate the dilation ($\oplus$) of T by $M_t$: $T_1 = T \oplus M_t = \{z \in E \mid ({M_t}^s)_z \cap T \neq \emptyset\}$, where E is a Euclidean space or an integer grid, ${M_t}^s = \{x \in E \mid -x \in M_t\}$, and $({M_t}^s)_z$ is the translation of ${M_t}^s$ by the vector z, i.e.,$({M_t}^s)_z = \{b + z \mid b \in M_t\}, \forall z \in E$;
2: Calculate the erosion ($\ominus$) of $T_1$ by $M_t$:$T_2 = T_1 \ominus M_t = \left\{z \in E \mid M_{t_z} \subseteq T_1\right\}$, where $M_{t_z}$ is the translation of $M_t$ by the vector z;
-*Morphological opening operation*-
3: Calculate the erosion of $T_2$ by $M_t$: $T_3 = T_2 \ominus M_t$;
4: Calculate the dilation of $T_3$ by $M_t$: $T_{ult} = T_3 \oplus M_t$;
```

However, DBSCAN does not consider temporal relation among the data points and hence cannot guarantee the continuity of the lane-change maneuver. To eliminate the noise of the labeled time series, a morphological operation [18] is applied to the dataset after DBSCAN clustering, as described in Algorithm 1. For example, in this study, we apply $M_t = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{1\times 5}$, as we assume the lane-change maneuver is continuous in a short period and at least lasts for 0.5 seconds with an update rate of 10 Hz.

B. Lane-Change Decision Prediction Based on LSTM

Before planning the trajectory, the human driver first considers high-level decisions, e.g., lane change and lane keep. In order to further analyze the trajectory detail, we first recognize the driver's intention. Therefore, in this study, the lane-change decision prediction is formulated as a time-series classification problem, which predicts the states in future time steps into either lane change or lane keep.

As a special recurrent neural network (RNN), LSTM network can model long-term temporal dependencies among time series [19], and it has proved its capability on the time-series prediction [20] [21]. Our goal is to classify future T steps actions $A_{t:t+T}$ into $\{a_{\text{change}}, a_{\text{keep}}\}$ given historical vehicle states. Since each vehicle state in the time series is highly correlated with its neighbors, the sequence-to-sequence LSTM network is adopted for a multi-step and multi-variable prediction.

The structure of our network is shown in Fig. 2. The neural network input is a sequence $\xi=\left(s_{t-T+1}^{lstm},\ldots,s_t^{lstm}\right)$ of the last T steps selected vehicle states, including the lateral deviation from the centerline of the lane (d_t^{lat}), lateral speed (v_t^{lat}), longitudinal speed (v_t^{lon}), and remaining distance for (mandatory) lane change (d_t^{remain} , if any). The output is the next T steps predicted lane-change action sequence

$(A_{t+1}, \ldots, A_{t+T+1})$. The network consists of two LSTM layers (each followed by one dropout layer) and two fully connected layers (with ReLu and Softmax layer as their activation layers). The labeled dataset is split into the training set, validation set, and test set. Since the units of vehicle state features are different, the whole dataset is normalized to the range of [0,1].

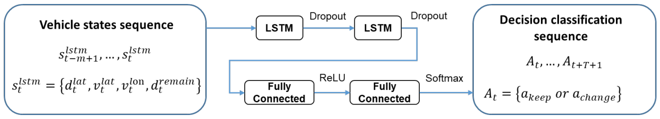

Vehicle states sequence
$s_{t-m+1}^{lstm}, \ldots, s_t^{lstm}$
$s_t^{lstm} = \{d_t^{lat}, v_t^{lat}, v_t^{\mathrm{lon}}, d_t^{remain}\}$
LSTM
Dropout
LSTM
Dropout
Fully Connected
ReLU
Fully Connected
Softmax
Decision classification sequence
$A_t, \ldots, A_{t+T+1}$
$A_t = \{a_{keep} \text{ or } a_{change}\}$

Fig. 2. Structure of the proposed neural network for lane-change decision prediction.

C. Probability Estimation and Trajectory Prediction Based on IRL

The driver behavior model is usually described by the cost function, which incorporates the rationality of the human driver and the theory of mind [14], and rational drivers behave by optimizing their cost function. Considering the continuity of the trajectory space, this study adopts Continuous IRL with Locally Optimal Examples [22] [16] to recover this unknown cost function from expert demonstrations.

1) Continuous IRL: The cost function is a linear combination of a set of features, i.e., $C_i(\theta_i, \xi) = \theta_i^T f_i(\xi), i = a_{\text{change}}, a_{\text{keep}}$, where θ_i^T is the weights vector emphasizing the features, and $f_i(\xi) = \|f_i(s_1, s_2, \ldots, s_t)\|_2$. The goal of the IRL is to figure out the θ_i^* of each driver, maximizing the likelihood of the driver's historical trajectories $\Xi = \{\xi_k\}, k = 1, \ldots, K$, shown in Equation (1):

$$\theta_i^* = \arg\max_{\theta_i} P(\Xi \mid \theta_i) \tag{1}$$

According to the principle of maximum entropy, as shown in Equation (2), a trajectory with a low cost has a higher probability, which is proportional to the exponential of its cost.

$$P(\xi \mid \theta_i) = \frac{e^{-C_i(\theta_i, \xi)}}{Z(\theta)} = \frac{e^{-\theta_i^T f_i(\xi)}}{\int e^{-\theta_i^T f_i(\tilde{\xi})} d\tilde{\xi}} \tag{2}$$

where $Z(\theta) = \int e^{-C_i(\theta_i, \tilde{\xi})} d\tilde{\xi}$ is the partition function integrating all arbitrary trajectories $\tilde{\xi}$. To handle the computational complexity in solving the partition function, the continuous IRL [22] [16] approximates the cost of arbitrary trajectory $C_i\left(\theta_i, \tilde{\xi}\right)$ using the second-order Taylor expansion around the demonstrated trajectory ξ, as in Equation (3). As a result, the partition function is now a Gaussian integral and becomes analytically solvable.

$$C_i\left(\theta_i, \tilde{\xi}\right) \approx C_i(\theta_i, \xi) + (\tilde{\xi} - \xi)^T \frac{\partial C}{\partial \xi} + (\tilde{\xi} - \xi)^T \frac{\partial^2 C}{\partial \xi^2} (\tilde{\xi} - \xi) \tag{3}$$

Then combining this approximation with equations (1) and (2), as in Equation (4), the problem can be reformed as minimizing the log-likelihood of $-\log P(\Xi \mid \theta_i)$.

$$\begin{aligned} \theta_i^* = \arg\min_{\theta_i} \sum_{k=1}^{K} & \frac{1}{2} \mathbf{g}_{\theta_i}^T(\xi_k) \mathbf{H}_{\theta_i}^{-1}(\xi_k) \mathbf{g}_{\theta_i}(\xi_k) \\ & - \frac{1}{2} \log |\mathbf{H}_{\theta_i}(\xi_k)| \end{aligned} \tag{4}$$

where $\mathbf{g}^T$ and $\mathbf{H}$ are the gradient and hessian, respectively. This formula indicates that along the expert demonstration the recovery cost function should have small gradients and large positive Hessians.

2) Cost function feature selection: The selected features present the vehicle state in an interpretable way and can capture the preference of the driver. We select the following features to calculate the cost function.

a) Car-following risk: the time headway to the leading vehicle,

$$f_{\text{risk }f} = 1 - \tanh(h_{ev}/H_{min}), h_{ev} = d_{headway}/v_{lon} \tag{5}$$

where $H_{\min}$ is the minimum safe time headway based on the 3-second rule [23], h_{ev} is the time headway of ego vehicle to leading vehicle, and $d_{headway}$ is the distance to the leading vehicle.

b) Lane-change risk: ego vehicle is projected to its adjacent lane and calculates the time headway to its potential leading vehicle and from its following vehicle,

$$f_{risk_{lc}} = \frac{1 - \tanh(h'_{ev}/H_{\min}) + 1 - \tanh\left(h'_{fv}/H_{\min}\right)}{2} \tag{6}$$

where $h_{ev}^{'}$ is the project time headway of ego vehicle to potential leading vehicle, and $h_{fv}^{'}$ is the time headway from the potential following vehicle.

c) Lane-change urgency: If ego vehicle needs to perform a mandatory lane change, the remaining distance should be considered.

$$f_{urge} = \frac{\left[1 + \tanh\left(\frac{L_{\text{width}}}{2} - y\right)\right]\left[1 + \tanh\left(\frac{(X_m - x)}{(v_{lon} \cdot H_{\min})}\right)\right]}{\max(f_{\text{urge}})} \tag{7}$$

where the L_{width} is the width of the lane, X_m is the longitudinal location of the midpoint of merging area, x and y are the locations of ego vehicle, and the $max(f_{urge})$ is the maximum of f_{urge}, for normalizing the feature.

An example surface of this feature is shown in Fig. 3, which illustrates how this feature varies within a 200m lane-change area, with a lane width of 4m. As the vehicle comes closer to the end of the lane-change area without changing the lane, the urgency increases. But once the lane change is completed, the urgency will decrease to zero shortly.

d) Mobility: Drivers have different preferences on mobility, and the difference between current speed and the speed limit (v_{lim}) is used to evaluate this preference.

$$f_m = 1 - e^{-(v_{lim} - v_{lon})^2} \tag{8}$$

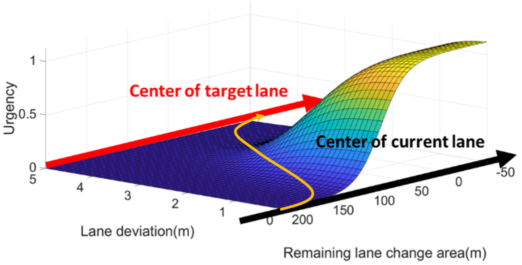


Fig. 3. The urgency for a mandatory lane change.

e) Comfort: The absolute value of the longitudinal acceleration a_{lon} and the lateral acceleration a_{lat} is used to gauge comfort preference.

$$f_{c1} = |a_{lon}|\,, f_{c2} = |a_{lat}| \tag{9}$$

f) Lane deviation: We also include lateral distance into the cost function to address the imperfection of driving along the centerline of the lane even in the lane-keep stage.

$$f_d = |y - Y_c| \tag{10}$$

where Y_c is the location of the centerline of the lane, y is the lateral position of the ego vehicle.

Considering the driver's focus may be different in each scenario (e.g., a driver may care about the lane-change risk and the remaining distance when changing the lane, but not when keeping the lane), we select two groups of features: $\left\{f_{risk_f}, f_{\text{risk}_{lc}}, f_{\text{urge}}, f_m, f_{c1}\right\}$ for lane-change maneuvers, and $\left\{f_{risk_f}, f_m, f_{c1}, f_{c2}, f_d\right\}$ for lane-keep maneuvers, respectively.

3) Trajectory evaluation: To execute the decision of lane change or lane keep, planning the trajectory is essential. Considering the real-time performance, instead of exploring arbitrary trajectory, we adopt a polynomial trajectory generator [15] to plan the candidate trajectories $\widetilde{\xi}_k$. As the trajectory generator block in Fig. 1 presents, at each time step, this trajectory generator takes the vehicle state $\{x, y, v, a\}$ as inputs and generates multiple trajectories within a prediction window. In this study, we set the time window as 4 seconds.

The cost function $C_i\left(\theta_i, \xi\right) = \theta_i^T f_i(\xi)$ is used to evaluate the probability of each possible trajectory $\widetilde{\xi}_k$, based on Equation (11). Considering the interaction and driver preference, the probability estimation for lane-change decision prediction ($\hat{a}_i = \hat{a}_{\text{keep}}, \hat{a}_{\text{change}}$) is evaluated by Equation (12), e.g., the probability of lane change equals the sum of the probability of all sampled lane-change trajectories.

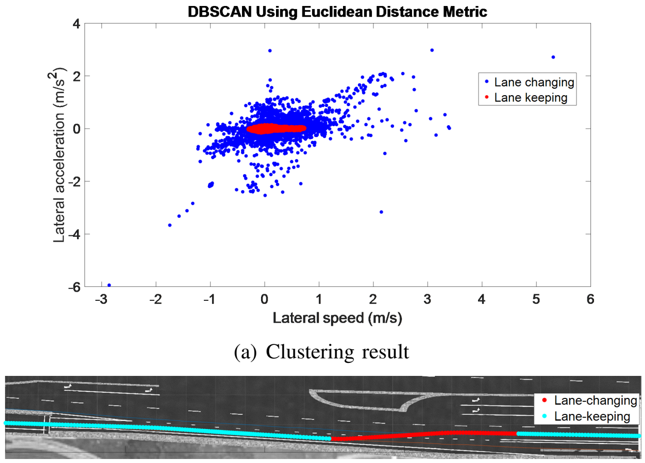
$$P\left(\widetilde{\xi}_k \mid \theta_i^*\right) = \frac{e^{-C_i\left(\theta_i{}^* \widetilde{\xi}_k\right)}}{\sum_{k=1}^{K} e^{-c_i\left(\theta_i^*, \widetilde{\xi}_k\right)}} \tag{11}$$

$$P\left(\hat{a}_i\right) = \sum_{k=1}^{K} P\left(\widetilde{\xi}_k \mid \theta_i{}^*\right) \tag{12}$$

IV. Experiments and Results

The proposed algorithm is designed on a Unity-SUMO co-simulation platform, which is built based on a Windows desktop, and a Logitech G27 Racing Wheel [24]. In this platform, a real-world traffic network is programmed in the Unity game engine, spanning from the intersection of Chicago Avenue to the intersection of Iowa Avenue along Columbia Avenue in Riverside, California. Traffic flow is generated by SUMO, and human input is consumed by the Logitech driving set. This platform allows various drivers to conduct human-in-the-loop simulations in an immersive traffic environment, where the driver data is collected, and the algorithm online validation is performed [25].

In this study, 37 trips with lane changes and 22 trips without any lane change within the on-ramp/off-ramp area are collected. The average duration of each trip is 30 seconds, with an update rate of 10 Hz. This data is processed by DBSCAN as shown in Fig. 4(a), and an example of labeled trajectory is shown in Fig. 4(b), indicating that the lane-change segment is well labeled.



(a) Clustering result

(b) Data labeling for each time step

Fig. 4. Trajectory labeling based on DBSCAN.

TABLE I
Recovered Feature Values for Different Types of Demonstrations

	f_{risk_f}	$f_{risk_{lc}}$	f_{urge}	f_m	f_{c1}	f_{c2}	f_d
$\theta_{change}{}^*$	0.276	0.278	0.334	0.047	0.054	-	-
$\theta_{keep}{}^*$	0.081	-	-	0.012	0.074	0.599	0.235

The recovered cost functions for two types of demonstrations are shown in Table I. For the lane-change behavior, the risk-related features and lane-change urgency contribute the most. In the lane-keep scenario, the features of lateral comfort and lane deviation are more important than other features.

As presented in Fig. 1, the online prediction process combines decision prediction and trajectory prediction. At each time step, the current vehicle state is sent into the LSTM network for decision prediction, where look-backward and prediction windows are both 3 seconds (30-time steps). The decision prediction result guides the system to the corresponding cost function which will be used to evaluate all the trajectory candidates. Based on Equation (11), we select the most probable trajectory as our prediction, i.e., $\hat{\xi} = \max_{\widetilde{\xi_k}} \left(P\left(\widetilde{\xi_k} \mid \theta_i{}^*\right)\right)$, and it can be projected into the simulation platform as shown in Fig. 5(a), where the probability of lane change is also estimated based on Equation (12).

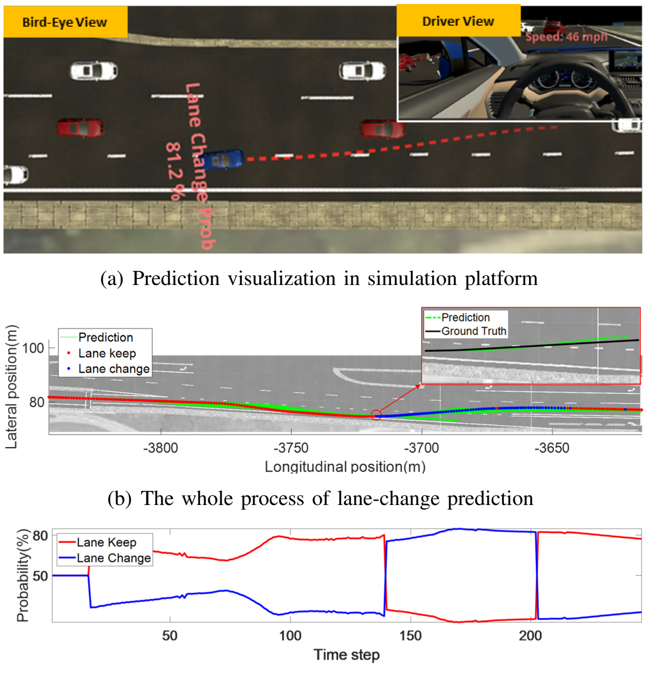
Fig. 5(b) presents the whole prediction process of the same trip of Fig. 5(a). Each time step of the ground-truth trajectory is labeled by LSTM in real time as lane keep (red dots) or lane change (blue dots). This prediction result shows that the lane-change decision is recognized in 3 seconds (30-time steps) before the vehicle crosses the borderline. Also, the visualized comparison of the 4-second horizon predicted trajectory (in green dash line) with the ground truth is shown in the zoom-in subfigure. Fig. 5(c) depicts the probability estimation of lane change and lane keep during a trip containing a lane-change maneuver, reflecting the intention of the driver. In addition, Fig. 6 displays a trip without any lane-change behavior, and the confidence of the prediction increases as the vehicle gets closer to the end of the lane-change area.

To quantify the accuracy of the predicted trajectories, the Mean Euclidean Distance (MED) [26] is adopted in this study. At time step t, the predicted trajectory $\hat{\xi}_t(L) = \left\{\widehat{xy}_t, ..., \widehat{xy}_{t+L}\right\}$ is compared with the ground truth $\xi_t(L) = \left\{xy_t, ..., xy_{t+L}\right\}$ within the same horizon L and the same sampling rate, as shown in Equation (13):

$$m_{\text{MED}}\left(\hat{\xi}_t, \xi_t\right) = \frac{1}{L}\sum_{l=1}^{L}\left\|\widehat{xy}_{t+l} - xy_{t+l}\right\|_2 \qquad (13)$$

where $xy_t = (x_t, y_t)$.

In our online test, the mean MED of 10 trips of the proposed method achieves 0.39 m with a 4-second prediction window. This result outperforms the IRL-based prediction method proposed by Sun et al. [16], which achieves a mean MED of 0.62 m with a 3 seconds prediction window. More importantly, we validate our methodology in an online fashion that allows drivers to test the system in human-in-the-loop simulations, while most other literature only did this in an offline fashion by running numerical simulations.

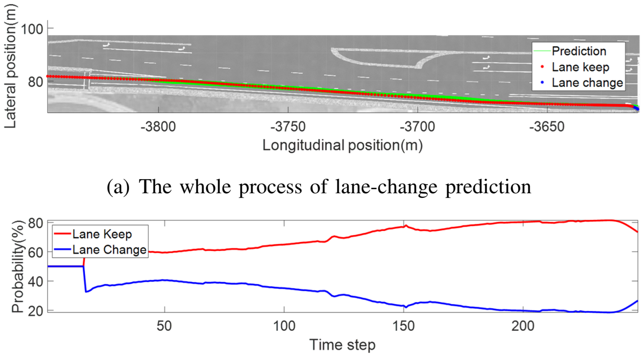


(a) Prediction visualization in simulation platform

(b) The whole process of lane-change prediction

(c) Predicted Lane change probability over time

Fig. 5. Online lane-change prediction of a trip with one lane change event.



(a) The whole process of lane-change prediction

(b) Predicted Lane change probability over time

Fig. 6. Online lane-change prediction of a trip without any lane change event.

V. Conclusion and Future Work

In this paper, we have proposed an LSTM-based online lane-change decision prediction method supported by an enhanced unsupervised data labeling method. We have also developed an online lane-change trajectory prediction along with driver preference recovery considering real-time vehicle states, vehicular interaction, and driving preference. Finally, we have performed the online algorithm validation in a mandatory lane-change scenario created in a human-in-the-loop simulation platform. The result has been quantified by the MED between the predicted trajectory and ground truth, showing an accuracy of 0.39 meters within a 4-second prediction window.

A major future step of this study is to recruit various drivers to conduct more comprehensive tests on the human-in-the-loop co-simulation platform and build a personalized driving model for each driver based on his or her personal dataset. This will help deepen the understanding of driver behaviors and further improve the prediction accuracy.

References

[1] Z. Wang, Y. Bian, S. E. Shladover, G. Wu, S. E. Li, and M. J. Barth, "A survey on cooperative longitudinal motion control of multiple connected and automated vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 1, pp. 4–24, 2020.

[2] Z. Wang, G. Wu, and M. J. Barth, "A review on cooperative adaptive cruise control (CACC) systems: Architectures, controls, and applications," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 2884–2891.

[3] Y. Xing, C. Lv, H. Wang, H. Wang, Y. Ai, D. Cao, E. Velenis, and F.-Y. Wang, "Driver lane change intention inference for intelligent vehicles: framework, survey, and challenges," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4377–4390, 2019.

[4] K. Schmidt, M. Beggiato, K. H. Hoffmann, and J. F. Krems, "A mathematical model for predicting lane changes using the steering wheel angle," *Journal of safety research*, vol. 49, pp. 85–e1, 2014.

[5] H. Berndt, J. Emmert, and K. Dietmayer, "Continuous driver intention recognition with hidden markov models," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2008, pp. 1189–1194.

[6] F. Li, W. Wang, G. Feng, and W. Guo, "Driving intention inference based on dynamic bayesian networks," in *Practical applications of intelligent systems*. Springer, 2014, pp. 1109–1119.

[7] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, A. Wedel, and W. Rosenstiel, "Object-oriented bayesian networks for detection of lane change maneuvers," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 3, pp. 19–31, 2012.

[8] H. Woo, Y. Ji, H. Kono, Y. Tamura, Y. Kuroda, T. Sugano, Y. Yamamoto, A. Yamashita, and H. Asama, "Dynamic potential-model-based feature for lane change prediction," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 000 838–000 843.

[9] Z. Shou, Z. Wang, K. Han, Y. Liu, P. Tiwari, and X. Di, "Long-term prediction of lane change maneuver through a multilayer perceptron," in *IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2020.

[10] J. Ashesh, H. Koppula, S. Soh, B. Raghavan, A. Singh, and A. Saxena, "Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture," *arXiv preprint arXiv:1601.00740*, 2016.

[11] V. Mahajan, C. Katrakazas, and C. Antoniou, "Prediction of lane-changing maneuvers with automatic labeling and deep learning," *Transportation research record*, vol. 2674, no. 7, pp. 336–347, 2020.

[12] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning." in *AAAI*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[13] Y. Hu, L. Sun, and M. Tomizuka, "Generic prediction architecture considering both rational and irrational driving behaviors," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3539–3546.

[14] M. Naumann, L. Sun, W. Zhan, and M. Tomizuka, "Analyzing the suitability of cost functions for explaining and imitating human driving behavior based on inverse reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5481–5487.

[15] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[16] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2111–2117.

[17] Y. Liu, Z. Wang, K. Han, Z. Shou, P. Tiwari, and J. Hansen, "Vision-cloud data fusion for adas: A lane change prediction case study," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2021.

[18] R. Van Den Boomgaard and R. Van Balen, "Methods for fast morphological image transforms using bitmapped binary images," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 3, pp. 252–258, 1992.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[21] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with KNN and LSTM," *Journal of Advanced Transportation*, vol. 2019, 2019.

[22] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," *arXiv preprint arXiv:1206.4617*, 2012.

[23] C. D. Handbook. (2021) Vehicle positioning. [Online]. Available: https://www.dmv.ca.gov/portal/handbook/california-driver-handbook/vehicle-positioning/

[24] X. Liao, X. Zhao, Z. Wang, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "Game theory-based ramp merging for mixed traffic with unity-sumo co-simulation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2021.

[25] X. Zhao, X. Liao, Z. Wang, G. Wu, M. J. Barth, K. Han, and P. Tiwari, "Co-simulation platform for modeling and evaluating connected and automated vehicles and human behavior in mixed traffic," *SAE MobilityRxiv™ Preprint*, 2021.

[26] J. Quehl, H. Hu, Ö. Ş. Taş, E. Rehder, and M. Lauer, "How good is my prediction? Finding a similarity measure for trajectory prediction evaluation," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.