# A Flexible and Explainable Vehicle Motion Prediction and Inference Framework Combining Semi-Supervised AOG and ST-LSTM

Shengzhe Dai, Zhiheng Li, *Member, IEEE*, Li Li, *Fellow, IEEE*, Nanning Zheng, *Fellow, IEEE*, and Shuofeng Wang

***Abstract*— Accurate trajectory prediction of surrounding vehicles is important for automated vehicles. To solve several existing problems of maneuver-based trajectory prediction, we propose four targeted solutions and establish a trajectory prediction model that integrates semi-supervised And-or Graph (AOG) and Spatio-temporal LSTM (ST-LSTM). To reduce the dependence on the well-labeled dataset, we introduce the concept of sub-maneuvers to improve the classifications of vehicle movements based on the given rough maneuver labels. AOG is used as the backbone of the probabilistic motion inference considering sub-maneuvers. We only define the basic units and inference logics of AOG and design a semi-supervised approach to directly learn the sub-maneuvers and the inference model structure from the training data, without manually specifying the structure (layers and nodes) of the inference model. This approach helps to avoid excessive artificial design or biases. The learned hierarchical motion inference model improves the interpretability of the overall trajectory prediction process. To utilize vehicle interaction information and further yield more accurate prediction, we adopt two different methods to consider vehicle interaction in the two sub-models (maneuver recognition and trajectory prediction). The experiment on NGSIM I-80 dataset shows that the maneuver-based model proposed in this paper (AOG-ST and refined AOG-ST-TB) performs more accurate trajectory prediction results. Although the AOG-ST seems clumsy and slow, we show that it is a flexible and quick model for trajectory prediction for various driving scenarios through the discussion and experiment.**

***Index Terms*— Trajectory prediction, maneuver recognition, maneuver-based model, and-or graph, semi-supervised learning, vehicle interactions.**

Manuscript received November 5, 2019; revised March 29, 2020 and June 8, 2020; accepted July 30, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101402, in part by the Science and Technology Innovation Committee of Shenzhen under Grant JCYJ20170818092931604, and in part by the Intel Collaborative Research Institute for Intelligent and Automated Connected Vehicles (ICRI-IACV). The Associate Editor for this article was L. Li. *(Corresponding author: Li Li.)*

Shengzhe Dai and Shuofeng Wang are with the Department of Automation, Tsinghua University, Beijing 100084, China.

Zhiheng Li is with the Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China, and also with the Department of Automation, Tsinghua University, Beijing 100084, China.

Li Li is with the Department of Automation, BNRist, Tsinghua University, Beijing 100084, China (e-mail: li-li@tsinghua.edu.cn).

Nanning Zheng is with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China.

Digital Object Identifier 10.1109/TITS.2020.3016304

## I. Introduction

TRAJECTORY prediction of surrounding vehicles is a basic function of automated vehicles [1]. It provides important information to support the subsequent motion planning and control of automated vehicles [2], [3] and further improve driving safety. The key problem of trajectory prediction is to identify the target vehicle's current maneuver and predict its future trajectory. Here, a maneuver is defined as a set of movements with similar semantic goals/operation rules. Such maneuvers can be intuitively understood and classified by human drivers (e.g., going straight maneuver, lane change maneuver, etc.). Some researchers also use the term "behavior" to refer to the same thing [1].

It should be pointed out that the frequently used term, "motion planning", covers a noticeably larger area of research topics; see discussion in [2], [3]. If we focus on a trajectory, motion planning refers to trajectory planning which generates a reference trajectory and makes the target vehicle to move along this trajectory. In short, motion prediction means to foresee other vehicles' trajectory in the near future, and motion planning means to guide itself trajectory in the near future.

As shown in [4], we can summarize most existing trajectory prediction models into two categories: **maneuver-based models** [1], [5]–[7] and **end-to-end models** [8]–[10]. Their difference lies in whether the prediction process contains an intermediate step of maneuver recognition. In other words, maneuver-based models first recognize the current maneuver and then perform trajectory prediction according to the recognition results. These two sequential steps are usually handled in different models. In contrast, end-to-end models skip the maneuver recognition step and directly generate prediction results.

This difference makes two kinds of models face different difficulties.

(1) *The difficulty of obtaining well-labeled dataset*

Maneuver-based models require well-labeled datasets to supervise their maneuver inference part. However, in practice, accurate data labeling is time-consuming and expensive. Resorting to rough labeled data may not help, since it will cause some similar but actually different movements (e.g. free lane change and forced lane change) to be classified into the same category. In addition, manual labeling errors

are usually hard to detect and may harm the performance of trained models.

In contrast, end-to-end models do not need explicitly labeled data to supervise their training. However, we have to manually filter out the incorrect trajectories when the training data contains lots of noise. Otherwise, incorrect data may noticeably affect the learning result.

(2) *The difficulty of model interpretability*

End-to-end models put the whole prediction process into a "black box" and thus lack of interpretability. Differently, maneuver-based models usually set the pre-defined maneuver category to be in agreement with human intuitions. The corresponding statistical inference of vehicle maneuver further enhances the interpretability of the prediction process.

(3) *The difficulty of model design and training*

Maneuver-based models decompose complex movements into several concise maneuvers that are relatively easy to be modeled by simple trajectory models [1]. However, we need to carefully design the probabilistic inference logics for maneuver inference part to avoid inappropriate artificial design or biases. Till now, we still do not have a mature design yet.

In contrast, end-to-end models aim to automatically identify various trajectories by using some complex enough deep neural network models. This solution simplifies the design problem but often makes the prediction model hard to train. As discussed in [4], we must carefully balance the proportion of different types of trajectories in the training set to eliminate the influence of imbalanced data. Otherwise, the trained model may neglect some important maneuvers that occur infrequently. For example, an incorrectly trained model may view any behaviors as going straight, since most datasets are dominated by going straight trajectory records.

(4) *The difficulty of using vehicle spatial interaction information*

Early trajectory prediction models only considered the ego historical information of the target vehicle. Some recent works [5], [11], [12] began to consider **vehicle spatial interactions** to yield more accurate trajectory predictions in dense traffic. These approaches can be further divided into two types:

The first type extracts some handcrafted features from the original trajectories of the target vehicle and interested surrounding vehicles to express vehicle interactions [5], [13]–[16]. The independent feature extraction process does not change the structure of subsequent models, nor does it increase the computational complexity, thus it can be directly combined with both maneuver-based models and end-to-end models as an extra input module. The corresponding feature extraction methods (e.g. potential field method [17]–[19], etc.) are designed based on our prior knowledge on the impact of vehicle interactions, and are thus interpretable. However, how to embed a prior knowledge remains to be further investigated.

The second type uses some complex learning models [4], [11], [20]–[23] to automatically discover features and directly learn the impact of surrounding vehicles on the target vehicle's trajectory. However, these learning models are usually complex for model design and training. In the recent studies, this kind of methods is usually used in end-to-end models and the second part (trajectory prediction) of maneuver-based models.

In this paper, we adopt the **maneuver-based** framework to support motion inference and yield more accurate predictions. To address the four difficulties of maneuver-based models, we propose a new trajectory prediction model that contains four targeted solutions.

First, we introduce the concept of sub-maneuvers to improve the classification of vehicle movements and further reduce the negative effect of rough maneuver labeling. The sub-maneuvers are learned in a semi-supervised approach based on incremental clustering.

Second, we use And-or Graph model [24], [25] as a backbone to organize hierarchical probabilistic motion inference based on learned sub-maneuvers and bind each sub-maneuver with a specific trajectory model. Compared to some conventional maneuver recognition models (such as SVM [26], [27], random forest classifier [28], HMM [29], DBN [7], RNN (includes LSTM) [5], [23], etc.) And-or Graph (AOG) is evaluated to be powerful to organize elaborate probability inference [24], [25]. Moreover, AOG model also improves the interpretability of trajectory prediction.

Third, we combine a dynamic maintenance algorithm with the semi-supervised learning approach to automatically adjust the structure and scale of the motion inference model to fit the training data. This approach helps to avoid excessive artificially designs on specific inference logics and make better use of data.

Fourth, we combine different methods to describe the impact of vehicle interactions in the two sub-models. The first method is used in maneuver recognition step to extract interaction features and keep the interpretability of probabilistic motion inference. The second method is ST-LSTM [4] which builds trajectory model for each learned sub-maneuver and yield more accurate predictions.

Fig. 1 summarizes the design principles, the corresponding solving models, and the prediction procedures proposed in this paper. We train and evaluate our overall model (denoted as AOG-ST) on the NGSIM I-80 dataset. AOG-ST (and the refined AOG-ST-TB) is compared with our previous end-to-end model (denoted as ST-LSTM) [4]. The results show that AOG-ST yields better prediction accuracy and explainability than ST-LSTM. We will also prove that AOG-ST is a flexible and quick trajectory prediction model for various driving scenarios, although it seems complex and slow. Finally, we discuss the sensitivity of AOG to motion changes. Although much recent work has not considered this sensitivity, we believe that it is a key to on-time forewarning and needs further study.

To give a detailed explanation, the rest of this paper is arranged as follows. *Section II* presents the problem we study, declares the basic notations of this paper and presents the calculation process of the overall model. *Section III* describes the structure and inference algorithm of our AOG model. *Section IV* introduces the initialization and dynamic maintenance algorithm of our AOG. *Section V* introduces the feature

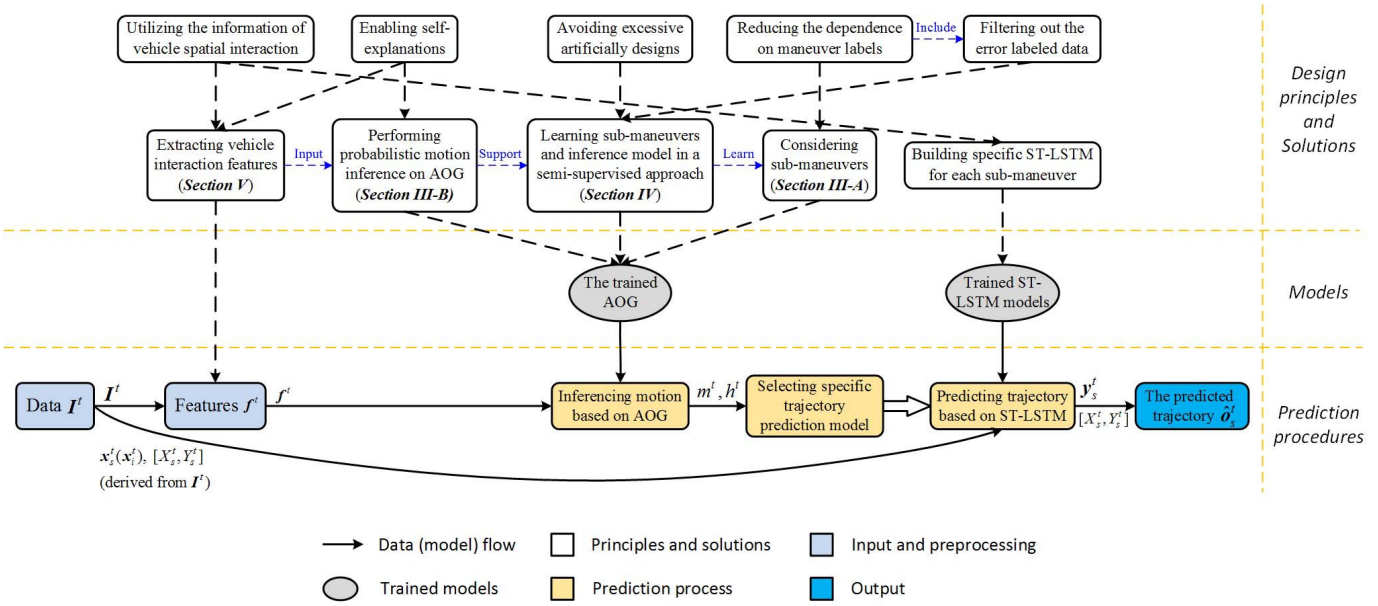



Fig. 1. The flowchart of our model.

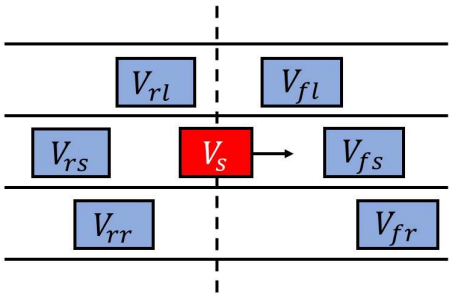



Fig. 2. The illustration of the target vehicle and its surrounding vehicles.

extraction method for AOG. *Section VI* presents the process of the experiment and discusses the results. Finally, *Section VII* concludes the paper.

## II. Problem Formulation

### A. The Input and Output of Prediction

The nomenclatures used in this paper are listed in *Appendix-B*.

As shown in [4] and Fig. 2, we need to study the target vehicle $V_s$ and six closest surrounding vehicles $V_{fl}$, $V_{rl}$, $V_{fs}$, $V_{rs}$, $V_{fr}$ and $V_{rr}$ (namely, the front-left, rear-left, front, rear, front-right and rear-right) respectively. Each surrounding vehicle is with a longitudinal distance of less than 80 $m$ from $V_s$.

The input $\boldsymbol{I}^t$ of our model includes the historical velocity and position sequences of the seven vehicles at time $t$. The outputs are the motion inference result $\{m^t, h^t\}$ and the predicted trajectory $\hat{\boldsymbol{o}}_s^t$ of $V_s$. Here, $m^t$ and $h^t$ respectively represent the most-likely maneuver and sub-maneuver of $V_s$, whose categories are learned by AOG.

Specially, the input is:

$$\boldsymbol{I}^t = \left[\boldsymbol{i}^{t-t_h}, \ldots, \boldsymbol{i}^{t-\Delta t}, \boldsymbol{i}^t\right]^T \tag{1}$$

$$\boldsymbol{i}^t = \left[\boldsymbol{i}_s^t, \boldsymbol{i}_{fl}^t, \boldsymbol{i}_{rl}^t, \boldsymbol{i}_{fs}^t, \boldsymbol{i}_{rs}^t, \boldsymbol{i}_{fr}^t, \boldsymbol{i}_{rr}^t\right] \tag{2}$$

where $t_h$ defines the length of historical time horizon and $\Delta t$ defines the time resolution level. $\boldsymbol{i}_s^t$ ($\boldsymbol{i}_i^t$) denotes the observation sequence of $V_s$ ($V_i$) at time $t$. Here, $i$ represents the index of surrounding vehicles, i.e., $i \in \{fl, rl, fs, rs, fr, rr\}$. $\boldsymbol{i}_s^t$ ($\boldsymbol{i}_i^t$) is a four-dimensional vector:

$$\begin{cases} \boldsymbol{i}_s^t = \left[v_{X,s}^t, v_{Y,s}^t, X_s^t, Y_s^t\right] \\ \boldsymbol{i}_i^t = \left[v_{X,i}^t, v_{Y,i}^t, X_i^t, Y_i^t\right] \end{cases} \tag{3}$$

where $v_{X,s}^t$ ($v_{X,i}^t$), $v_{Y,s}^t$ ($v_{Y,i}^t$), $X_s^t$ ($X_i^t$) and $Y_s^t$ ($Y_i^t$) define the instantaneous lateral and longitudinal velocities, the lateral and longitudinal coordinates in the natural coordinates, respectively.

The output $\hat{\boldsymbol{o}}_s^t$ consists of a series of position coordinates:

$$\hat{\boldsymbol{o}}_s^t = \left[\hat{X}_s^{(t+\Delta t)}, \hat{Y}_s^{(t+\Delta t)}, \ldots, \hat{X}_s^{(t+t_p)}, \hat{Y}_s^{(t+t_p)}\right] \tag{4}$$

where $t_p$ defines the prediction time horizon. The length of the prediction sequence is $n_p = t_p/\Delta t$.

During training, the manually labeled maneuver $M^t$ and the ground truth future trajectory $\boldsymbol{o}_s^t$ of $V_s$ are used to supervise the training of AOG and ST-LSTM respectively.

$$\boldsymbol{o}_s^t = \left[X_s^{(t+\Delta t)}, Y_s^{(t+\Delta t)}, \ldots, X_s^{(t+t_p)}, Y_s^{(t+tp)}\right] \tag{5}$$

$M^t$ describes the initial maneuver clusters of the research scenario and serves as the basis of the semi-supervised AOG training. The AOG training algorithms and the application of $M^t$ are detailed in *Section IV*. The training of ST-LSTM minimizes the deviation between $\hat{\boldsymbol{o}}_s^t$ and $\boldsymbol{o}_s^t$. The detailed selection of the loss function is presented in [4].

### B. The Calculation Process

As shown in Fig. 1, given an input sequence $\boldsymbol{I}^t$, we need to implement the following steps to complete the motion inference and trajectory prediction.

First, we extract the feature vector $\boldsymbol{f}^t$ from the original input $\boldsymbol{I}^t$ for AOG. Second, we perform motion inference on AOG based on $\boldsymbol{f}^t$ and obtain the most-likely maneuver $m^t$ and sub-maneuver $h^t$ of $V_s$. Third, we select the most appropriate ST-LSTM trajectory prediction model based on the $m^t$ and $h^t$. Finally, we perform trajectory prediction based on the selected ST-LSTM and output the predicted trajectory $\hat{\boldsymbol{o}}_s^t$. The specific input, output and notations of each step are as follows.

*Step 1: Feature extraction*

The input of this step is the original input $\boldsymbol{I}^t$ (a $\frac{t_h+\Delta t}{\Delta t} \times 28$ matrix) and the output is the extracted feature vector $\boldsymbol{f}^t$ at time $t$:

$$\boldsymbol{f}^t = \left[f_1, \ldots, f_{n_f}\right] \tag{6}$$

where $n_f$ defines the dimension of $\boldsymbol{f}^t$.

The feature extraction process is a heuristic refinement of the original input $\boldsymbol{I}^t$. The extracted features (especially the vehicle interaction features) reflect our prior knowledge of vehicle motion, which simplify the motion inference calculation and improve the inference reliability. Note that the extracted features are only used for the motion-level inference, rather than the calculation of the predicted trajectory $\hat{\boldsymbol{o}}_s^t$. In the subsequent trajectory prediction step, the used vehicle interaction features are automatically learned in ST-LSTM models.

In this paper, we construct an eight-dimensional $\boldsymbol{f}^t$ that includes six ego features of the target vehicle and two features of spatial vehicle interactions. The specific feature extraction methods are proposed in *Section V*.

*Step 2: Motion inference based on AOG*

The input of this step is the extracted feature vector $\boldsymbol{f}^t$ and the output is the estimated maneuver $m^t$ and sub-maneuver $h^t$ of $V_s$. The probabilistic motion inference is organized on AOG.

*Step 3: Trajectory prediction model selection*

The input of this step is the estimated $m^t$ and $h^t$. In this paper, we bind each learned sub-maneuver category with a specific ST-LSTM trajectory prediction model to better train similar movements. According to $m^t$ and $h^t$, we select the corresponding ST-LSTM model for subsequent trajectory prediction.

*Step 4: Trajectory prediction*

The input of this step is the original input $\boldsymbol{I}^t$ and the output is the predicted trajectory (coordinate sequence) $\hat{\boldsymbol{o}}_s^t$ of $V_s$. The specific prediction process relies on the selected ST-LSTM models obtained in the previous step.

The core idea of ST-LSTM is to explicitly distinguish the temporal relation of vehicle trajectory segments and the spatial interaction effects on trajectories, which play different roles in the prediction process. ST-LSTM constructs two types of LSTM models: Component-LSTM models the temporal law of each vehicle without considering its surrounding vehicles and generates a preliminary prediction sequence. Interaction-LSTM models the spatial impact of each specific pair of surrounding vehicle and target vehicle and generates a correction sequence, which semantically means the adjustment of the target vehicle to avoid the safety risk of the corresponding surrounding vehicle. The risk of each surrounding vehicle to the target vehicle is evaluated to calculate the dynamic weight for the superimposition of the corresponding correction sequence. The superimposition of all the sequences above generates the final prediction result.

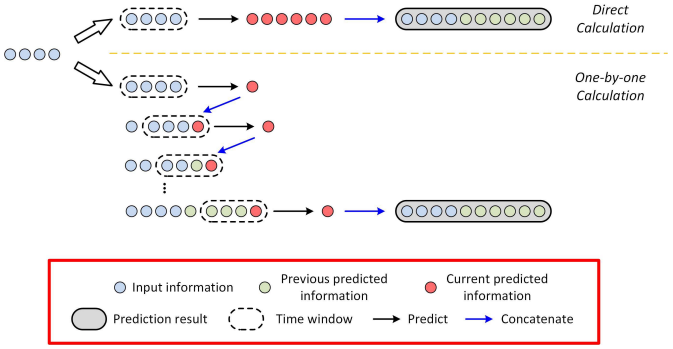



Fig. 3. The illustration of the two kinds of ways to execute trajectory predictions.

The specific definition and implementation of ST-LSTM refer to [4], here, we only introduce its input and output interface and consider it as a module. The input of ST-LSTM trajectory prediction model is the coordinate displacement sequence $\boldsymbol{x}_s^t$ ($\boldsymbol{x}_i^t$) of $V_s$ ($V_i$) and the output is the predicted coordinate displacement sequence $\boldsymbol{y}_s^t$ of $V_s$:

$$\begin{cases} \boldsymbol{x}_s^t = \left[\Delta X_s^{(t-t_h+\Delta t)}, \Delta Y_s^{(t-t_h+\Delta t)}, \ldots, \Delta X_s^t, \Delta Y_s^t\right] \\ \boldsymbol{x}_i^t = \left[\Delta X_i^{(t-t_h+\Delta t)}, \Delta Y_i^{(t-t_h+\Delta t)}, \ldots, \Delta X_i^t, \Delta Y_i^t\right] \end{cases} \tag{7}$$

$$\boldsymbol{y}_s^t = \left[\Delta \hat{X}_s^{(t+\Delta t)}, \Delta \hat{Y}_s^{(t+\Delta t)}, \ldots, \Delta \hat{X}_s^{(t+t_p)}, \Delta \hat{Y}_s^{(t+t_p)}\right] \tag{8}$$

where $\Delta X_s^t = X_s^t - X_s^{(t-\Delta t)}$ and $\Delta Y_s^t = Y_s^t - Y_s^{(t-\Delta t)}$ defines the **ground truth** lateral and longitudinal coordinate displacements from the time $(t-\Delta t)$ to $t$. $\Delta \hat{X}_s^{(t+\Delta t)} = \hat{X}_s^{(t+\Delta t)} - \hat{X}_s^t$ and $\Delta \hat{Y}_s^{(t+\Delta t)} = \hat{Y}_s^{(t+\Delta t)} - \hat{Y}_s^t$ defines the **predicted** lateral and longitudinal coordinate displacements from the time $t$ to $(t+\Delta t)$ (Here, $\hat{X}_s^t = X_s^t$ and $\hat{Y}_s^t = Y_s^t$). The length of the historical sequence $\boldsymbol{x}_s^t$ ($\boldsymbol{x}_i^t$) is $n_h = t_h/\Delta t$. The length of $\boldsymbol{y}_s^t$ is $n_p$.

$\boldsymbol{x}_s^t$ ($\boldsymbol{x}_i^t$) can be directly derived from $\boldsymbol{I}^t$. $\hat{\boldsymbol{o}}_s^t$ can be calculated by the cumulative sum given $\boldsymbol{y}_s^t$ and the cooridinate $\left[X_s^t, Y_s^t\right]$ of $V_s$ at time $t$.

Note that $\boldsymbol{y}_s^t$ is not always obtained in one calculation, specifically, it depends on the $n_p$ and the pre-defined output length $n_p'$ of ST-LSTM. We agree that there are two kinds of ways to execute trajectory predictions (short-term and long-term predictions), as shown in Fig. 3.

The first way is one-by-one calculation. Each step of the calculation only outputs one sequence point (i.e., $n_p' = 1 \ll n_p$). Therefore, $\boldsymbol{y}_s^t$ is iteratively calculated through the sliding window input. Based on this method, short-term predictions are part of long-term predictions. However, the number of

iterations between them may be over large, resulting in serious error accumulations. Moreover, it also faces the problem of time-consuming due to the iterations.

The second way is directly calculation. That is, we predetermine the dimension of output trajectories to be the same as the given prediction horizon (i.e., $n'_p = n_p$). Based on this method, short-term predictions and long-term predictions are simultaneously obtained. However, this will make the prediction model less flexible.

In this paper, we predefine a moderate $n'_p$ (i.e., $1 \ll n'_p < n_p$) to avoid the above problems. We perform a sliding window on the overall sequence (with a step width of $n'_p$) to generate input sequences and further predict a series of predicted sequence segments by ST-LSTM. The final $\boldsymbol{y}_s^t$ is generated by concatenating the above predicted sequences.

The above prediction process does not explicitly predict vehicle speeds and accelerations, because the adopted ST-LSTM is a data-driven model without modeling vehicle kinematics and dynamics. But our AOG framework is flexible enough to support the estimation of these two variables. The reason and explanation for these facts are listed as follows:

Due to the flexibility of our framework, we can enable the prediction part to explicitly predict speeds and accelerations through some appropriate modifications on the network of ST-LSTM. Even without the above modifications, we can calculate the speed and acceleration values through the difference of the obtained position coordinates of vehicles. However, due to the lack of restraint by vehicle dynamics or kinematics, these results may have large errors or even be unfeasible.

Moreover, a large number of previous work has proven that current state-of-the-art data-driven models have significantly higher prediction accuracy than classical kinematic/dynamic models [1]. This is partly because many simplifications of kinematic/dynamic models bring much error. Currently, researchers still do not find an appropriate way to combine data-driven methods and kinematic/dynamic models. We are studying how to reach this goal and hope to find a way in the future.

The AOG-based probabilistic motion inference (*Step 2*) is the core step of the overall trajectory prediction procedure. The training of the AOG structure and inference logics are also the core contribution of this paper. In the following two sections, we will present the details of the structure and probabilistic inference of AOG (*Section III*) and the semi-supervised training algorithm of AOG (*Section IV*).

## III. The AOG Motion Inference Model

The recent work [24] constructs a vehicle maneuver inference model in a single scenario based on AOG. In this paper, in order to consider multilevel driving behaviors and support hierarchical motion inference, we modify the structure of AOG in [24] and design the corresponding inference logics. Specifically, we add a sub-maneuver layer below the AOG in [24] to organize a more elaborate inference. To build a layer-by-layer inference logic and reduce computational complexity, we decouple the calculation of different layers by some relaxation operations.

In this section, we will introduce the detailed structure of our AOG model (*Section III-A*) and the probabilistic motion inference based on AOG (*Section III-B*). In *Section III-C*, we will introduce the calculation of state transition probabilities, which is an import part of motion inference calculation.

### A. Graphic Structure of Our Model

In this subsection, we will introduce different kinds of nodes in our AOG model and explain how to build a motion inference model by combining these nodes layer by layer. The overall structure of AOG is shown in Fig. 4, which is the unfolding drawing of the ellipse "The trained AOG" in Fig. 1.

*1) Or-Nodes:* As shown in the left part of Fig. 4, the front view of the AOG model is a tree-like structure with alternant And-node layers and Or-nodes layers. Here, each Or-node has a clear semantics, e.g. $S$ represents a specific scenario, $M$ represents a maneuver, and $H$ represents a sub-maneuver. The Or-nodes with the same parent node can convert to each other. Their state transition relationships can be described by a probability graph model, which is represented by the horizontal blue dotted lines in the left part of Fig. 4. We can formulate a series of ***state transition graphs (STG)*** at the bottom of AOG; see the blue blocks of Fig. 4. Each blue dotted arrow in STGs indicates one possible state transition, and the corresponding transition probability is determined by the And-node above it.

*2) And-Nodes:* Each And-node represents a configuration selected by its parent Or-node, which configures the parameters to calculate state transition probabilities of the STG below it. To facilitate the understanding, we can treat the configuration as the ***"style"*** of state transition.

Suppose we set two state transition styles: aggressive (state transitions are frequent and Logistic functions are steep) and conservative (state transitions are infrequent and Logistic functions are even). We set two collocated And-nodes in one AOG layer to express the alternative relationship between the two styles. When performing motion inference in this layer, we first evaluate the most-likely style based on historical inference results. Then, we select the corresponding set of parameters (of this style) to calculate the state transition probabilities below this And-node. When we finish the overall inference, the obtained inference result that uses these state transition probabilities is more in agreement with the current motion characteristic (style) of the target vehicle.

Note that there is no direct correlation between the styles (the number and parameters) under different Or-nodes. Ideally, we should get the optimal number of styles for each Or-nodes by clustering the frequency of state transitions. But for simplicity, we assume that the numbers of styles **under all the Or-nodes** are equal.

*3) STG Nodes and State Transition Probabilities:* We divide the nodes in STG into two types: trivial nodes and non-trivial nodes. It is easy to define trivial nodes based on our prior knowledge, e.g. straight road is the trivial scenario and going straight is the trivial maneuver in straight road scenarios. We use self-loops to represent trivial nodes in STGs, as shown in the first STG in Fig. 4. The state transition probability

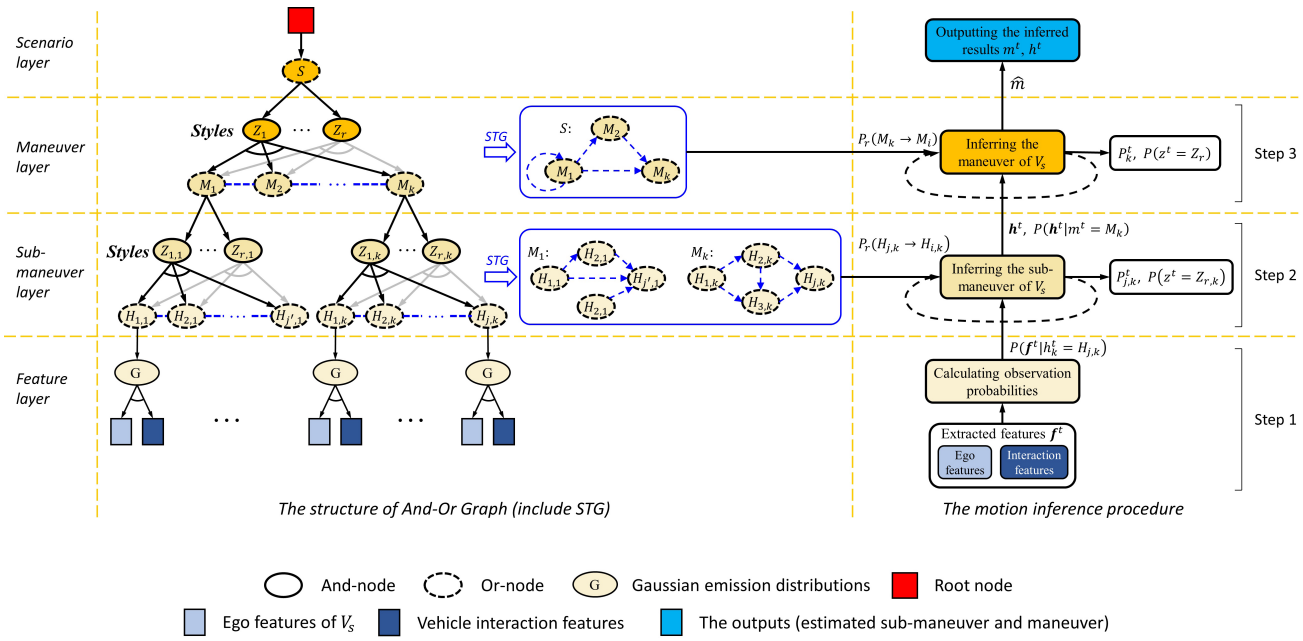



Fig. 4. The illustration of the hierarchical vehicle maneuver inference model based on AOG. The $Z_r$ and $Z_{r,k}$ in this figure respectively express the selected style of state transition (i.e., vehicle motion) in different layers. When changing styles, we switch the arrow group below the corresponding And-node (e.g., from the black arrow group to the gray arrow group below $Z_1$) to change the local inference logic of AOG.

from trivial nodes to others should be obtained by statistical methods.

In contrast, the state will eventually transit from non-trivial nodes to other nodes within a finite time. In this paper, we use Logistic regression (LR) of duration to calculate the state transition probability from non-trivial nodes to others [24]. The calculation process is described in *Section III-C*.

*4) The Layers of AOG:* As shown in Fig. 4, the AOG model contains four layers: scenario layer, maneuver layer, sub-maneuver layer and feature layer.

The feature layer serves as the input layer of AOG motion inference and calculates the emission probabilities.[1] The input feature vector contains ego features of the target vehicle and vehicle interaction features. In this paper, we assume that the input feature vector follows a multi-dimensional Gaussian distribution for each sub-maneuver. Therefore, we bind each considered sub-maneuver with a specific Gaussian emission distribution and build the connections between the feature layer and the sub-maneuver layer.

The other three layers serve as the inference layers that generate probabilistic estimates of semantics at different levels. Each layer consists of one And-node sublayer and one Or-node sublayer. We will detail the inference calculation of these layers in the following subsection. In this paper, we only consider the inference in a single research scenario. Thus the scenario layer in Fig. 4 only contains one And-node.

*5) Basic Notations of AOG:* As shown in Fig. 4, we assume the maneuvers (learned by AOG) in research scenario $S$ are $\{M_k|1 \le k \le n_k\}$ and the sub-maneuvers in maneuver $M_k$ are $\{H_{j,k}|1 \le j \le n_{j,k}\}$. For simplicity, we assume that each Or-node in the trained AOG has $N_{style}$ And-nodes (driving styles) below it. Therefore, the configurations belonging to $S$ and $M_k$ are denoted as $\{Z_r|1 \le r \le N_{style}\}$ and $\{Z_{r,k}|1 \le r \le N_{style}\}$ respectively

[1]We borrow this term from HMM [42] to describe the conditional probability of the observation (e.g. the feature vector $\boldsymbol{f}^t$) given a specific maneuver or sub-maneuver. The definition and usage of this term in this paper are the same as those in classical HMM models.

### B. Probabilistic Motion Inference Based on AOG

As shown in the right part of Fig. 4, the motion inference of AOG is a bottom-up process and contains three steps.

*Step 1: The calculation in the feature layer*

The input of the feature layer is the feature vector $\boldsymbol{f}^t$ extracted from the original input $\boldsymbol{I}^t$. Then, we can calculate the emission probability $P\left(\boldsymbol{f}^t|h_k^t = H_{j,k}\right)$ of each sub-maneuver $H_{j,k}$ by the corresponding Gaussian emission distribution.

*Step 2: The inference in the sub-maneuver layer*

The input of the sub-maneuver layer is the emission probability $P\left(\boldsymbol{f}^t|h_k^t = H_{j,k}\right)$ obtained from the feature layer. The output is the posterior probability $P_{j,k}^t = P\left(h_k^t = H_{j,k}|\boldsymbol{f}^t\right)$ of each $H_{j,k}$ subordinate to $M_k$. Moreover, the sub-maneuver layer providers the emission probability for the maneuver layer.

The sub-maneuver inference also need some internal inputs, including the posterior probability $P_{j,k}^{t-\Delta t}$ at the time $(t - \Delta t)$, the duration $d_{j,k}$ of each $H_{j,k}$ and the probability $P\left(z_k^{t-\Delta t} = Z_{r,k}\right)$ that $V_s$ adopts the style $Z_{r,k}$ up to time $(t - \Delta t)$.

Let us take the inference below a specific $M_k$ as an example. If the And-nodes (style configurations) are not considered, it is straightforward that $P_{i,k}^t$ is calculated by:

$$P_{i,k}^t \propto P\left(\boldsymbol{f}^t|h_k^t = H_{i,k}\right) \sum_j P\left(H_{j,k} \to H_{i,k}\right) P_{j,k}^{t-\Delta t} \quad (9)$$

where the state transition probability $P\left(H_{j,k} \rightarrow H_{i,k}\right)$ can be calculated from $d_{j,k}$ if $H_{j,k}$ is a non-trivial STG node; see explanations in *Section III-C*. Otherwise, $P\left(H_{j,k} \rightarrow H_{i,k}\right)$ is a constant value that learned by AOG.

Consider the state transition style $z_k^{t-\Delta t} \in \left\{Z_{r,k} | 1 \leq r \leq n_r\right\}$, the Eq.(9) becomes:

$$\begin{aligned} &P_r\left(h_k^t = H_{i,k} | \boldsymbol{f}^{t-\Delta t}\right) \\ &\quad = \sum_j P_r\left(H_{j,k} \rightarrow H_{i,k}\right) P_r\left(h_k^{t-\Delta t} = H_{j,k} | \boldsymbol{f}^{t-\Delta t}\right) \end{aligned} \tag{10}$$

$$\begin{aligned} &P_r\left(h_k^t = H_{i,k} | \boldsymbol{f}^t\right) \\ &\quad \propto P\left(\boldsymbol{f}^t | h_k^t = H_{i,k}\right) P_r\left(h_k^t = H_{i,k} | \boldsymbol{f}^{t-\Delta t}\right) \end{aligned} \tag{11}$$

$$P_{i,k}^t \propto \sum_r P\left(z_k^{t-\Delta t} = Z_{r,k}\right) P_r\left(h_k^t = H_{i,k} | \boldsymbol{f}^t\right) \tag{12}$$

where $P_r$ represents the probability under the state transition style $Z_{r,k}$. $P\left(z_k^{t-\Delta t} = Z_{r,k}\right)$ is the probability that the target vehicle adopts $Z_{r,k}$ up to the time $(t - \Delta t)$.

Finally, we determine the most-likely sub-maneuver **below** $M_k$ by selecting $\hat{h}_k^t = \underset{H_{i,k}}{\arg\max} P_{i,k}^t$. We concatenate $\hat{h}_k^t$ and the historical most-likely sub-maneuver sequence $\boldsymbol{h}_k^{t-\Delta t}$ and obtain $\boldsymbol{h}_k^t = \left[\boldsymbol{h}_k^{t-\Delta t}, \hat{h}_k^t\right]$. Here, $\boldsymbol{h}_k^t$ is used to update $P\left(z_k^t = Z_{r,k}\right)$:

$$\begin{aligned} P\left(z_k^t = Z_{r,k}\right) &= P\left(z_k^t = Z_{r,k} | \boldsymbol{h}_k^t\right) \propto P\left(\boldsymbol{h}_k^t | z_k^t = Z_{r,k}\right) \\ &= \prod_\tau P\left(\hat{h}_k^\tau | z_k^t = Z_{r,k}, \hat{h}_k^{\tau-\Delta t}, \hat{h}_k^{\tau-2\Delta t}, \ldots\right) \end{aligned} \tag{13}$$

Suppose $\hat{h}_k^\tau$ is just related to $\hat{h}_k^{\tau-\Delta t}$, Eq. (13) then becomes:

$$P\left(z_k^t = Z_{r,k}\right) \propto \prod_\tau P\left(\hat{h}_k^{\tau-\Delta t} \rightarrow \hat{h}_k^\tau\right) \tag{14}$$

We can traverse all the $M_k$ and obtain a candidate sub-maneuver vector $\boldsymbol{h}^t = \left[\hat{h}_1^t, \ldots, \hat{h}_k^t\right]^T$ that stores the most likely sub-maneuver below each maneuver of AOG. Note that we have not yet gotten the final $h^t$. The final determination of $h^t$ depends on the inference result of the maneuver layer.

To calculate the emission probability for the maneuver layer, we treat the inferred most-likely sub-maneuver $\hat{h}_k^t$ as an observation of its corresponding $M_k$ given $\boldsymbol{f}^t$. Therefore, the emission probability of $M_k$ is approximated as the emission probability of $\hat{h}_k^t$, i.e., $P\left(\boldsymbol{h}^t | m^t = M_k\right) \approx P\left(\boldsymbol{f}^t | h^t = \hat{h}_k^t\right)$. Based on this relaxation, all the emission probabilities of the maneuver layer can be directly derived from $P_{j,k}^t$, rather than by tracing back to $\boldsymbol{f}^t$ (the feature layer). i.e., the overall inference process can be organized layer-by-layer.

*Step 3: The inference in the maneuver layer*

The input of this layer is the candidate sub-maneuver vector $\boldsymbol{h}^t$ and the emission probability $P\left(\boldsymbol{h}^t | m^t = M_k\right)$ of each $M_k$. The output is the posterior probability $P_k^t = P\left(m^t = M_k | \boldsymbol{h}^t\right)$ that $V_s$ performs each $M_k$, the estimated maneuver $m^t$, and the estimated sub-maneuver $h^t$.

Similar to the inference in the sub-maneuver layer, some internal inputs are also needed in this layer, e.g., the posterior probability $P_k^{t-\Delta t}$ at the time $(t - \Delta t)$. The duration $d_k$, and the probability $P\left(z^t = Z_r\right)$ that $V_s$ adopts the style $Z_r$ up to time $(t - \Delta t)$.

Derived from Eq.(9), the maneuver inference regardless of state transition style $Z_r$ follows:

$$P_i^t \propto P\left(\boldsymbol{h}^t | m^t = M_i\right) \sum_k P\left(M_k \rightarrow M_i\right) P_k^{t-\Delta t} \tag{15}$$

Considering state transition style $z^{t-\Delta t} \in \{Z_r | 1 \leq r \leq n_r\}$, we can imitate Eq.(10)-(14) to implement the probabilistic maneuver inference and the update of $z^t$, finally obtain the most-likely maneuver $\hat{m} = \arg\max_{M_k} P_k^t = m^t$. Then the corresponding $\hat{h}_k^t$ is treated as the most-likely sub-maneuver $h^t$.

Our layer-by-layer inference can effectively reduce the computational complexity. Without the relaxation of the emission probability of maneuver, some "cross-maneuver" state transitions like $H_{j,k} \rightarrow H_{j,l}$ will be legal and need to consider, which will cause a great increase of state transition functions (and calculations). Moreover, to calculate the emission probability of $M_k$, we must trace down to all the sub-maneuvers (including sub-maneuvers of other maneuvers). This calculation is over complicated and hard to implement.

### C. The Calculation of State Transition Probability

In this paper, we adopt Logistic functions to calculate the state transition probabilities from non-trivial nodes given the duration of the last state (e.g. $P\left(H_{j,k} \rightarrow H_{i,k}\right)$ in Eq.(6)). Here, we introduce the calculation and the training method of these functions.

Let $h$ be an arbitrary non-trivial node of a certain STG. Denote $succ(h)$ as the set of all the successors of $h$, i.e., all nodes that can be converted from $h$. Let $f_{h \rightarrow h'}$ be the Logistic state transition probability function between $h$ and $h' \in succ(h)$, whose parameters are $\alpha_{h \rightarrow h'}$ and $\beta_{h \rightarrow h'}$. The probability of this transition is:

$$P\left(h \rightarrow h'\right) \propto \frac{1}{1 + \exp\left[-\left(\alpha_{h \rightarrow h'} d + \beta_{h \rightarrow h'}\right)\right]} \tag{16}$$

where $d$ is the duration of $h$.

The probability of keeping the state $h$ is:

$$P(h \rightarrow h) \propto \frac{1}{n} \sum_{\gamma \in \text{succ}(h)} \left(1 - P(h \rightarrow \gamma)\right) \tag{17}$$

where $n$ is the number of states in $succ(h)$.

To obtain the value of $\alpha_{h \rightarrow h'}$ and $\beta_{h \rightarrow h'}$, we use Logistic Regression to fit the Logistic function $f_{h \rightarrow h'}$. First, we traverse the training set and record the duration $d_i$ of keeping state $h$ before the $h \rightarrow h'$ transition occurs. The recorded dataset is denoted as $D = \{d_i | 1 \leq i \leq N_d\}$. Then we sample $N_{ds}$ duration points according to the range of $D$ and get sampled set $DS = \left\{ds_j | 1 \leq j \leq N_{ds}\right\}$. Given any data $d_i \in D$, a set of labeled training data can be derived:

$$T_i = \left\{\left(ds_j, \sigma_{i,j}\right) | 1 \leq j \leq N_{ds}\right\} \tag{18}$$

where $\sigma_{i,j} = \begin{cases} 1 \text{ if } d_i \leq ds_j \\ 0 \text{ else} \end{cases}$.

The likelihood of $D$ based on $f_{h\to h'}$ is:

$$\mathcal{L} = -\sum_{i=1}^{N_d}\sum_{j=1}^{N_{ds}}\left[\sigma_{i,j}\log f_{h\to h'}\left(ds_j\right) + \left(1-\sigma_{i,j}\right)\log\left[1-f_{h\to h'}\left(ds_j\right)\right]\right] \quad (19)$$

Then the parameters $\alpha_{h\to h'}$ and $\beta_{h\to h'}$ can be optimized based on gradient-based methods. In this paper, we have actually filtered out the abnormal data with error labels before starting the optimization. Otherwise, the abnormal data will have a big impact on the likelihood $\mathcal{L}$ and optimized results. We will describe the filtering approach of abnormal points during model training in *Section IV*.

## IV. Initialization And Dynamic Maintenance of AOG

In this section, we will present the initialization and dynamic maintenance method for AOG training.

(1) *Solving the difficulty of AOG initialization*

The core idea of semi-supervised AOG initialization is using a clustering method to classify adjacent points (with the same maneuver label) in the feature space into the same sub-maneuver. The main difficulty is how to filter out the abnormal points in feature space caused by error labeling. Since we assume that the feature vector follows a multi-dimensional Gaussian distribution for each sub-maneuver. If some abnormal points that are actually far away from all sub-maneuvers are forcibly divided into one sub-maneuver, the distribution of the sub-maneuver will be flatter and corresponding classification ability is weakened.

In this paper, we adopt DBSCAN as the core clustering method [30]. This density clustering algorithm can divide points at high-density areas in the feature space into clusters, while the outliers (at low-density area) will be well filtered out. The clustered sub-maneuvers serve as good complements to the rough maneuver labels and achieve a more elaborate division of vehicle motion.

(2) *Solving the difficulty of AOG dynamic maintenance*

After building AOG models, with the introduction of new training data, some new sub-maneuvers may be discovered or some sub-maneuvers could be connected in the feature space (i.e., merging to one sub-maneuver). Therefore, a dynamic maintenance method is needed to maintain that the sub-maneuver clusters always fit the amplified training set well.

The main difficulty is how to determine whether a sub-maneuver should split or some sub-maneuvers should aggregate. In this paper, we adopt the Bayesian Information Criterion (BIC) as the determination basis. The core idea is to find the better one from the original sub-maneuver (sub-maneuvers) and the split sub-maneuvers (merged sub-maneuver) that better fits the training set. Our dynamic maintenance algorithm will modify the clustered sub-maneuvers $\left\{H_{j,k}|1\le j\le n_{j,k}\right\}$ of each $M_k$ and the corresponding STG (e.g. $\alpha_{h\to h'}$ and $\beta_{h\to h'}$, etc.).

In this section, we will first introduce the two basic algorithms (DBSCAN and BIC) in *Section IV-A*. Then we will present the detailed specific initialization (*Section IV-B*) and dynamic maintenance (*Section IV-C*) approaches. The presented DBSCAN and BIC will be used in these two approaches.

### A. Classical Algorithms That Are Used

*1) Density Clustering Based on DBSCAN:* In this paper, we use the DBSCAN algorithm to perform density clustering on the feature space and obtain a preliminary sub-maneuver classification [30]. We use Euclidean distance as the distance measure of DBSCAN and use KD-tree to achieve faster nearest neighbor search. To implement better pre-classification, we select the parameters including distance threshold $\varepsilon$ and minimum number $n$ through experiments.

*2) Node Splitting and Merging Based on BIC:* In this paper, we propose node splitting and merging criterions based on the Bayesian Information Criterion (BIC) for dynamic maintenance of sub-maneuver classification [31]. The splitting criterion is used to determine whether one sub-maneuver in the AOG should be further divided into two sub-maneuvers. The merging criterion is used to determine whether two learned sub-maneuver is similar enough to be merged.

For simplicity, we assume the node splitting is one into two and the node merging is two into one. Then the splitting and merging problems are transformed into one specific problem that selecting one model which better fits the training set from two candidate models. Specifically, under the Gaussian distribution assumption of the feature vector, the two candidate models are respectively a single Gaussian model $G$ and a GMM $G'$ (mixed by $G_1$ and $G_2$).

Suppose the dataset for BIC decision is $F = \{f_i|1\le i\le N\}$. $\theta, \theta_1, \theta_2$ respectively denote the parameters of $G$, $G_1$, $G_2$. Then the BIC value of the two candidate models are:

$$BIC(G) = -2\sum_{i=1}^{N}\ln\left(p\left(f_i|\theta\right)\right) + d\ln N \quad (20)$$

$$BIC\left(G'\right) = -2\sum_{i=1}^{N}\ln\left(p_{\max}\left(f_i,\theta_1,\theta_2\right)\right) + d\ln N_1 + d\ln N_2 \quad (21)$$

where $p\left(f_i|\theta\right)$ represents the likelihood of the sample $f_i$ under the parameter $\theta$ (i.e., model $G$). d is the dimension of the input feature $f_i$. The $p_{\max}\left(f_i,\theta_1,\theta_2\right)$, $N_1$ and $N_2$ in Eq.(21) are calculated by:

$$\begin{cases} p_{\max}\left(f_i,\theta_1,\theta_2\right) = \max\left(p\left(f_i|\theta_1\right), p\left(f_i|\theta_2\right)\right) \\ N_1 = \sum_{i=1}^{N} 1\left\{p\left(f_i|\theta_1\right)\ge p\left(f_i|\theta_2\right)\right\} \\ N_2 = \sum_{i=1}^{N} 1\left\{p\left(f_i|\theta_1\right)\le p\left(f_i|\theta_2\right)\right\} \end{cases} \quad (22)$$

where $1\{a\} = \begin{cases} 1 & \text{if } a \text{ is True} \\ 0 & \text{else} \end{cases}$.

The first item of the BIC equation is the negative log-likelihood of the corresponding training set. The other items are penalty items, which are used to suppress the over-segmentation of the sub-maneuvers. A lower BIC value indicates that the model can better reflect the distribution characteristic of the dataset.

Suppose $H$ is the sub-maneuver to be decided for node splitting, which follows Gaussian distribution $G$. The feature set belongs to this sub-maneuver is $F$. The node splitting algorithm contains three steps:

1) Perform K-means clustering ($K = 2$) on the dataset $F'$ which is downsampled from $F$;
2) Label all the data of $F$ based on the K-means clustering result. Fit Gaussian distribution $G_1$ and $G_2$ based on the two categories of labeled data;
3) Calculate the BIC values of $G$ and $G'$ based on Eq.(20)-(22). Split $H$ into $H_1$ and $H_2$ which respectively follow $G_1$ and $G_2$ if $BIC(G)$ and $BIC(G')$ satisfy:

$$\frac{BIC\left(G'\right)}{BIC(G)} < Th_s \tag{23}$$

where $Th_s$ is the threshold of node splitting ($Th_s < 1$).

Suppose $H_1$ and $H_2$ are the sub-maneuvers to be decided for node merging, which follow Gaussian distribution $G_1$ and $G_2$ respectively. The feature set belongs to the sub-maneuvers are $F_1$ and $F_2$. The node merging algorithm contains two steps:

1) Fit Gaussian distribution $G$ based on $F = F_1 \cup F_2$;
2) Calculate the BIC values of $G$ and $G'$ based on Eq.(20)-(22). Merge $H_1$ and $H_2$ into $H$ which follows $G$ if $BIC(G)$ and $BIC(G')$ satisfy:

$$\frac{BIC\left(G'\right)}{BIC(G)} > Th_m \tag{24}$$

where $Th_m$ is the threshold of node merging. $Th_m$ should be larger than $Th_s$ to avoid oscillations between splitting and merging.

In the subsequent experiments, when selecting the values of $Th_s$ and $Th_m$, we test multiple sets of these parameters and choose a better set that can maintain a moderate frequency of node splitting and merging.

### *B. AOG Initialization Algorithm*

We design a semi-supervised approach to initialize AOG from the labeled training set. First, we use the maneuver labels to supervise the establishment of the maneuver layer. Then we perform the DBSCAN clustering algorithm (*Section IV-A*) on each maneuver node to learn sub-maneuvers. During the clustering process, some outliers that resulting from error labeling can be filtered out, thus the model can reduce the impact of error labeling on sub-maneuver clustering results. Finally, we perform node splitting (*Section IV-A*) on the clustered sub-maneuver to ensure that the number of sub-maneuver clusters in the learned AOG is above a certain number.

The specific initializing procedure is detailed as follows and also shown in *Appendix-C* (**Algorithm 1**). The main input of this algorithm is the labeled training set and the output is the initialized AOG.

1) Construct the scenario layer and maneuver layer according to the known scenario and maneuver labels. Add an additional unclassified model on each layer to store data without the corresponding label.
2) Traverse the training set and store each data into the maneuver nodes according to its label.
3) Perform DBSCAN-based pre-classification algorithm on each maneuver node. Construct preliminary sub-maneuver nodes after filtering out the noise classes.
4) Repeatedly traverse the preliminary sub-maneuvers (of each maneuver) and perform node splitting on each node, until no sub-maneuver can be split or the total number of the sub-maneuvers reaches a specified upper value. Then generate the STG among these sub-maneuvers.
5) Finally, we generate the STG among all the maneuvers.

The STG generating algorithm used above is described as follows and also shown in *Appendix-C* (**Algorithm 2**). The main inputs of the algorithm are the set of patterns with state transition relationships and the corresponding training set. The output is the STG among these patterns.

1) Traverse the training set and inference the pattern of each data based on the emission probability. When a state transition is captured, we record the state transition pair and the duration of the old state.
2) Traverse all the state transition pairs. If the transition is from a non-trivial node, we cluster the duration sequence (the number of clusters is equal to the preset number of state transition styles) and fit the state transition functions based on LR. If the transition is from a trivial node, we set the transition probability of this state transition pair to a constant value.

### *C. AOG Dynamic Maintenance Algorithm*

In this subsection, we will propose the dynamic maintenance algorithm of AOG. Given the initialized AOG and some new training data, the algorithm repeated executes the BIC node merging and splitting algorithm (*Section IV-A*) to fit the amplified training set.

The specific maintenance procedure is detailed as follows and also shown in *Appendix-C* (**Algorithm 3**). The main input of this algorithm is the new labeled training set and the output is the updated AOG model.

1) Traverse the new training set. We store the data with new maneuver labels and built corresponding sub-maneuvers by **Algorithm 1** when enough data is accumulated. If the maneuver label is already in AOG, we inference the sub-maneuver of the data based on the method in *Section III-B*, store the data in the most-likely sub-maneuver node and update the sub-maneuver when enough data is accumulated.
2) To update a sub-maneuver $h$, we first update the Gaussian distribution for observation and the state transition probability functions of related directed edges in STG. Then we traverse other sub-maneuvers (below the same maneuver) and perform node merging with $h$. Finally, we perform node splitting.

## V. Feature Extraction for AOG

As mentioned in *Section II-B*, the feature vector used in AOG includes **ego features** that are derived from the historical trajectory of the target vehicle and **interaction features** that are derived from the vehicle interaction relationships. In this section, we will present the detailed feature extraction method for AOG. Ego feature extraction is presented in *Section V-A*, and interaction feature extraction is presented in *Section V-B,C*.

### A. Ego Features of the Target Vehicle

Many conventional works adopt some statistics (e.g. RMS, mean, etc.) of vehicle velocity (coordinate) in specific historical time horizons to express the characteristics or changes of the vehicle motion in the recent period of time [32]–[34]. These extracted features have been validated to be effective in vehicle motion inference.

In this paper, we adopt **six ego features**: the instantaneous velocity in longitudinal and lateral directions, the mean and RMS value of lateral velocity in the historical horizon of 1 $s$ and 2 $s$. Note that these ego features need to be normalized before inputting to AOG.

### B. Vehicle Interaction Features Obtained From Potential Field

Compared to ego features, interaction features are usually heuristic and incorporate our prior knowledge. Some works adopt risk assessment indicators as interaction features, such as time to collision (TTC) and time to lane (TTL) [35]. The potential field method is also a commonly used approach to measure the risk level [17]–[19]. Although the values of potential functions do not have clear physical meanings, the construction of the potential field heuristically simulates the interactions between vehicles.

Some other works attempt to extract more abstract semantic features as the measure of vehicle interactions [14], [36], e.g., the prior probability that the target vehicle performs some specific motions given the current traffic situation. This kind of features is more interpretable and also harder to design.

In this paper, we present a combined interaction feature extraction method. We first adopt a potential field method to extract preliminary interaction features. Then we conduct further feature extraction based on our prior knowledge and obtain two features. These two features approximate the prior left and right lane change probabilities affected by the surrounding vehicles (without considering the ego historical movement of the target vehicle). In this subsection, we will present the potential field-based feature extraction method. We will present the further feature extraction method in *Section V-C*.

The used potential field method is derived from the concept of safety risks, here, we detail the method as follows.

*1) One-Dimensional Safety Gap:* In the longitudinal direction (along the road), we measure the safe distance by the formula of safe gap proposed in [37]. Let $V_l$ ($V_f$) be the leading (following) vehicle between $V_i$ and $V_s$. The safe distance between the two vehicles is:

$$D_y = v_f \rho + \frac{(v_f)^2 - (v_l)^2}{2a_{brake}} + L = \frac{\bar{v}\,\Delta v}{a_{brake}} + (v_f \rho + L) \quad (25)$$

where $v_l$ ($v_f$) is the longitudinal velocity of the $V_l$ ($V_f$). $L$ is the average length of the two vehicles. $\rho$ is the mean response time of drivers. $a_{brake}$ is the brake deceleration of the two vehicles. The average velocity $\bar{v} = (v_l + v_f)/2$. The relative velocity $\Delta v = v_f - v_l$.

This safe distance imagines a car-following scenario, the leading vehicle brakes by $a_{brake}$ until a full stop, and the following vehicle keeps uniform during the response time $\rho$, and then brakes by $a_{brake}$ until a full stop. This formula can be considered as (a lower bound of) safe distance because when two vehicles are closer than $D$, the following vehicle may not be able to avoid collision by braking.

In the lateral direction (vertical to the road direction), we set the safety distance to half of the lane width $L_{lane}$:

$$D_x = \frac{L_{lane}}{2} \quad (26)$$

*2) Two-Dimensional Safety Field:* We can roughly measure the safety status between two vehicles in two directions by comparing the actual distances to the safety distances ($D_y$ and $D_x$). However, the evaluation in the longitudinal and lateral directions are independent. If two cases have different "safe value" in both of the two directions, it is hard to compare their safety status.

In this paper, we build a two-dimensional safety field and link the longitudinal and lateral measures by equipotential lines; see Fig. 5. We assume that the equipotential lines of the safety field are concentric ellipses. In other words, the longitudinal and lateral safety gaps are equipotential ($A$ and $B$ in Fig. 5).

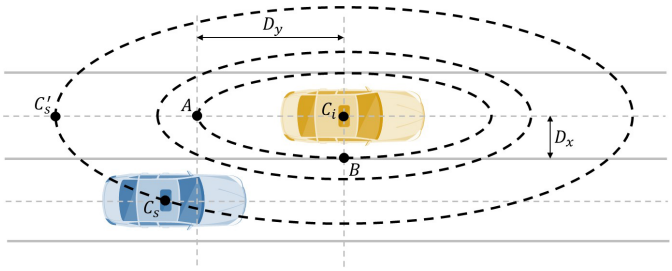



Fig. 5. The illustration of safety field.

Suppose the coordinates of $V_s$ and $V_i$ are $C_s = [X_s, Y_s]^T$ and $C_i = [X_i, Y_i]^T$ respectively. $\Theta_i$ is the safety potential field generated by $V_i$.

Based on the assumption of elliptical equipotential lines, $C_s$ is equipotential with $C_s'$, which has the same lateral coordinate with $C_i$. The longitudinal distance between $C_s'$ and $C_i$ is:

$$r = \sigma_y \sqrt{(C_s - C_i)^T \, \Sigma^{-1} \, (C_s - C_i)} \quad (27)$$

where $\Sigma = \mathrm{diag}\left(\sigma_x^2, \sigma_y^2\right)$. Here, $\sigma_x = D_x$ and $\sigma_y = D_y$.

Therefore, the two-dimensional safety potential field problem is transformed into a one-dimensional potential function

problem. In this paper, we adopt Eq.(28) as the potential function. There are still some conventional works adopt other functions (such as the density function of 2-D Gaussian distribution). We will discuss the advantage of our potential functions over 2-D Gaussian distribution in *Appendix-A*.

$$\vartheta_i = \Theta_i\left(C_s'\right) = \Theta_i\left(C_s\right) = A\frac{\exp(-r)}{r} \tag{28}$$

The potential function above lacks a numerical calibration, thus $\vartheta_i$ generated from different $V_i$ cannot be superimposed directly. Here, we calibrate the potentials by calculating the constant $A$ of Eq.(28):

$$\Theta_i\left(\left[\sigma_x, \sigma_y\right]^T\right) = 1 \tag{29}$$

The potential value above depicts the safety risk of $V_i$ to $V_s$. A larger potential value indicates that $V_i$ could have a more significant effect on the future motion of $V_s$. Then we can calculate the potential value of each $V_i$ ($i \in \{fl, rl, fs, rs, fr, rr\}$).

### C. Further Feature Extraction Based on Prior Knowledge

In this subsection, we perform further feature extraction based on the extracted $\vartheta_i$ in *Section V-B* and generate two new features. The new features incorporate our prior knowledge of the vehicle movements in the research scenario, which helps to reduce the feature dimension and accelerates the learning process. Specifically, the two features have the semantic of the prior left and right lane change probabilities.

Take the scenario of straight roads as an example, in this paper, we superimpose the potentials of the same lane to express the driving condition of the three lanes.

$$\begin{cases} \varphi_s &= \left(\vartheta_{fs}\right)^{\beta_s}\left(\vartheta_{rs}\right)^{(1-\beta_s)} \\ \varphi_l &= \left(\vartheta_{fl}\right)^{\beta_l}\left(\vartheta_{rs}\right)^{(1-\beta_l)} \\ \varphi_r &= \left(\vartheta_{fr}\right)^{\beta r}\left(\vartheta_{rr}\right)^{(1-\beta_r)} \end{cases} \tag{30}$$

where $\beta_s, \beta_l, \beta_r \in [0, 1]$. We use this form of superposition is to obtain a sum formula when calculating log-likelihood.

Here, a larger $\varphi$ indicates that the lane has a greater repulsion on $V_s$, i.e., approaching the lane may increase the security risk of $V_s$.

According to our prior knowledge, we respectively use $\ln\varphi_s/\varphi_l$ ($\ln\varphi_s/\varphi_r$) and $\ln\varphi_s$ to represent the subjective will and objective condition of lane changes. A large $\ln\varphi_s/\varphi_l$ ($\ln\varphi_s/\varphi_r$) indicates that the driving condition of the left (right) lane is superior to the current lane, which may attract the driver to perform lane change. A large $\ln\varphi_s$ indicates that $V_{fs}$ and $V_{rs}$ are very close to $V_s$, i.e., $V_s$ has limited lane change condition, thus the lane change probability is relatively low.

The reprocessed features $\ln\varphi_s/\varphi_l$ ($\ln\varphi_s/\varphi_r$) and $\ln\varphi_s$ have more interpretability than $\vartheta_i$, thus we use these features to model the prior left and right lane change probabilities based on Logistic functions:

$$\begin{cases} P_{rlc} = \dfrac{1}{1+e^{-g_r(\boldsymbol{\vartheta})}} \\ g_r(\boldsymbol{\vartheta}) = \alpha_{r1}\ln\dfrac{\varphi_s}{\varphi_r} - \alpha_{r2}\ln\varphi_s \end{cases} \tag{31}$$

$$\begin{cases} P_{llc} = \dfrac{1}{1+e^{-g_l(\boldsymbol{\vartheta})}} \\ g_l(\boldsymbol{\vartheta}) = \alpha_{l1}\ln\dfrac{\varphi_s}{\varphi_l} - \alpha_{l2}\ln\varphi_s \end{cases} \tag{32}$$

where $P_{llc}$ ($P_{rlc}$) is the prior left (right) lane change probability. $\alpha_{r1}, \alpha_{r2}, \alpha_{l1}, \alpha_{l2} \in [0, 1]$. Without loss of generality, we can set $\alpha_{r_1} + \alpha_{r_2} = 1$ and $\alpha_{l1} + \alpha_{l2} = 1$.

Then the parameters $(\beta_s, \beta_l, \beta_r, \alpha_{r1}, \alpha_{r2}, \alpha_{l1}, \alpha_{l2})$ could be optimized by Maximum Likelihood Estimation (MLE) and the Frank-Wolfe Algorithm. The optimized parameters are shown in Table I. The evaluations of $P_{rlc}$ and $P_{llc}$ are plotted in Fig. 6.

In Fig. 6, the x-coordinate represents the prior lane change probability just considering the traffic conditions, i.e., vehicle interactions. The y-coordinates represents the actual lane change frequency corresponding to each value of prior lane change probability. The positive correlation between x and y coordinates indicates that $P_{llc}$ and $P_{rlc}$ can provide reasonable prior judgments on vehicle maneuver based on vehicle interactions regardless of its historical movement. Therefore, we select $P_{llc}$ and $P_{rlc}$ as the vehicle interaction features for AOG. Since the Logistic function in Eq. (31)-(32) can limit the feature values in [0,1], it is no need to normalize these features.

Finally, we concatenate the six ego features extracted in *Section V-A* with the two interaction features extracted in *Section V-C* and construct an 8-dimensional feature vector for AOG. (i.e., the lateral and longitudinal instantaneous velocity,

TABLE I
THE OPTIMIZED PARAMETERS

| Parameters | $P_{llc}$ | $P_{rlc}$ |
|---|---|---|
| $\alpha_{l1}$ ($\alpha_{r1}$) | 0.838 | 0.844 |
| $\alpha_{l2}$ ($\alpha_{r2}$) | 0.162 | 0.156 |
| $\beta_s$ | 0.405 | 0.317 |
| $\beta_l$ ($\beta_r$) | 0.425 | 0.464 |

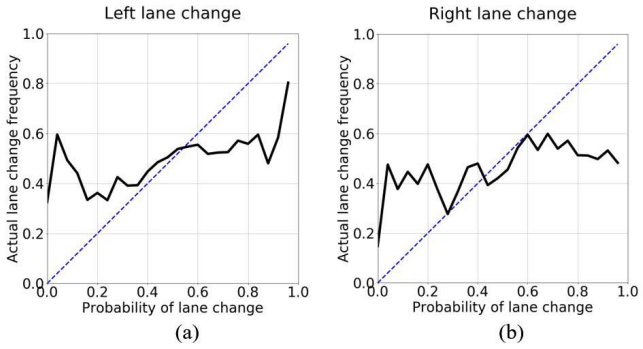



Fig. 6. The comparison between prior lane change probability and statistical lane change frequency: (a) Left lane change, (b) Right lane change.

the mean and RMS value of lateral velocity in the historical horizon of 1 s and 2 s, $P_{llc}$ and $P_{rlc}$).

## VI. Experiments

### A. Dataset and Experiment Settings

*1) Dataset Preprocessing:* We train and evaluate our model on the NGSIM I-80 dataset [43]. I-80 was collected on Interstate 80 in Emeryville, California, which is a straight road with multiple lanes. This dataset provides a wealth of trajectory information including lateral and longitudinal position and velocity, lane ID, the vehicle length and width, etc. However, it does not provide the boundary coordinates between lanes. Moreover, the dataset is unbalanced: Although it includes a large number of lane change operations, going straight trajectory still dominates the dataset, and the left (right) lane changes are also unbalanced.

In order to compare our maneuver-based model with our previous end-to-end model (ST-LSTM) [4], we adopt the dataset preprocessing method proposed in [4], including dataset balancing and trajectory trimming, to obtain the same training and testing set. To enable the quick search of the surrounding vehicles, we also perform necessary preprocessing that records the neighboring relations between vehicles according to their positions.

Moreover, in order to support the initialization algorithm of AOG, we manually label the maneuver categories for all trajectories in the preprocessed dataset. Limited by the difficulty of maneuver labeling, we just provide five types of rough category labels: going straight (GS), left lane change (LLC), merging into the left lane (MLL), right lane change (RLC), merge into the right lane (MRL).

Our maneuver labeling method for each specific trajectory is described as follows. Since I-80 dataset provides the lane ID of each vehicle at each sampling time, it is straightforward to know whether the trajectory contains lane changes (and the number of lane change). If no lane change is found, we can label all the time of the trajectory as GS, otherwise, we mark the jump time of lane ID as the boundary between LLC and MLL (RLC and MRL). Then we artificially determine the start and end time of each lane change by observing the trajectory curve, which is respectively marked as the boundary between GS and LLC (RLC) and the boundary between MLL (MRL) and GS. Finally, we set the maneuver label for each moment of the trajectory.

*2) Training Details:* Our training process is step-by-step. First, we initialize the AOG by a subset of the training set. Then we traverse the training set and dynamically maintain the AOG. After obtaining the final AOG, we bind each sub-maneuver with a specific ST-LSTM model. Finally, we train all the ST-LSTM on the training set.

Since our AOG initialization algorithm includes performing DBSCAN-based pre-classification on high dimensional feature space, if the dataset for initialization is too large, the neighbor searching for DBSCAN is time-consuming. Therefore, we just use a subset of the training set during initialization to speed up the calculation. We have tested some proportions and finally select 10% of the training set for initialization.

During the AOG initialization and dynamic maintenance, we set $N_{sub} = 8$ (**Algorithm 1**) to prevent excessive segmentation of sub-maneuvers. The number of styles $N_{style} = 2$ (**Algorithm 2**). $N_{buf}$ should be determined according to the size of the training set. In this paper, we set $N_{buf} = 100$. The threshold of AOG node splitting and merging are respectively $Th_s = 0.8$ and $Th_m = 0.9$. The parameters to calculate safe distance are $a_{brake} = 6\ m/s^2$, $\rho = 0.5\ s$ and $L_{lane} = 4\ m$. The hyperparameters $\beta_s, \beta_l, \beta_r, \alpha_{r1}, \alpha_{r2}, \alpha_{l1}, \alpha_{l2}$ are calculated in *Section V-C*.

As proposed in [4], the training of ST-LSTM contains two sub-steps. The training of $C_s(C_i)$ and the training of $I_i$. $C_s(C_i)$ and $I_i$ are all modeled by LSTM, respectively depicting the temporal relations of vehicle trajectories and the influence of vehicle spatial interactions on the trajectory of $V_s$. In this paper, we train a specific $C_s(C_i)$ for each sub-maneuver to learn its own trajectory characteristics. We share the $I_i$ between LLC and MLL (RLC and MRL) to reduce calculation, however, GS has its own $I_i$ because the influence of the surrounding vehicles during going straight is totally different from that during changing lanes. The detailed training process and loss function of ST-LSTMs are proposed in [4].

Although we set buffers in **Algorithm 3** to avoid too few similar points to aggregate into a separate category (sub-maneuver), it is not guaranteed that all ST-LSTM models get enough trajectory data for training. We have discussed the critical volume of ST-LSTM training set in [4], and we properly relax this requirement in this paper. We check all the ST-LSTMs after training and delete the insufficient trained ones due to limited data (and the corresponding nodes in AOG).

*3) Compared Models:* In this paper, we will compare the RMS prediction error between the following models: the end-to-end model **ST-LSTM** (i.e., the ST-LSTM-1350 in [4]), the maneuver-based model based on AOG and ST-LSTM (denoted as **AOG-ST**) and one refined model **AOG-ST-TB**, which is generated by trimming some bad clustered sub-maneuvers from AOG-ST. Here, we directly compared with our previous ST-LSTM model because it is evaluated to have a lower prediction error than some SOTA models in [4].

### B. The RMS Testing Results and Some Examples

In the following subsections, we will discuss the accuracy of our model (AOG-ST and AOG-ST-TB) from **two aspects**. In *Section VI-B*, we perform a basic numerical comparison of the models and demonstrate the characteristic that our model is sensitive to motion changes through several prediction examples. In *Section VI-C*, we focus on the "false alarm" predictions that are caused by the sensitivity to motion changes, and discuss the impact of false alarms on the prediction error.

In this paper, we adopt the RMS prediction error as the error metric [4], [11]. Here, the RMS prediction error (denoted as $RMS^P$) given prediction horizon $P$ follows:

$$ms^P = \frac{1}{P}\sum_{i=1}^{P}\left(\boldsymbol{\sigma}_i^t\right)^T \boldsymbol{\sigma}_i^t \tag{33}$$

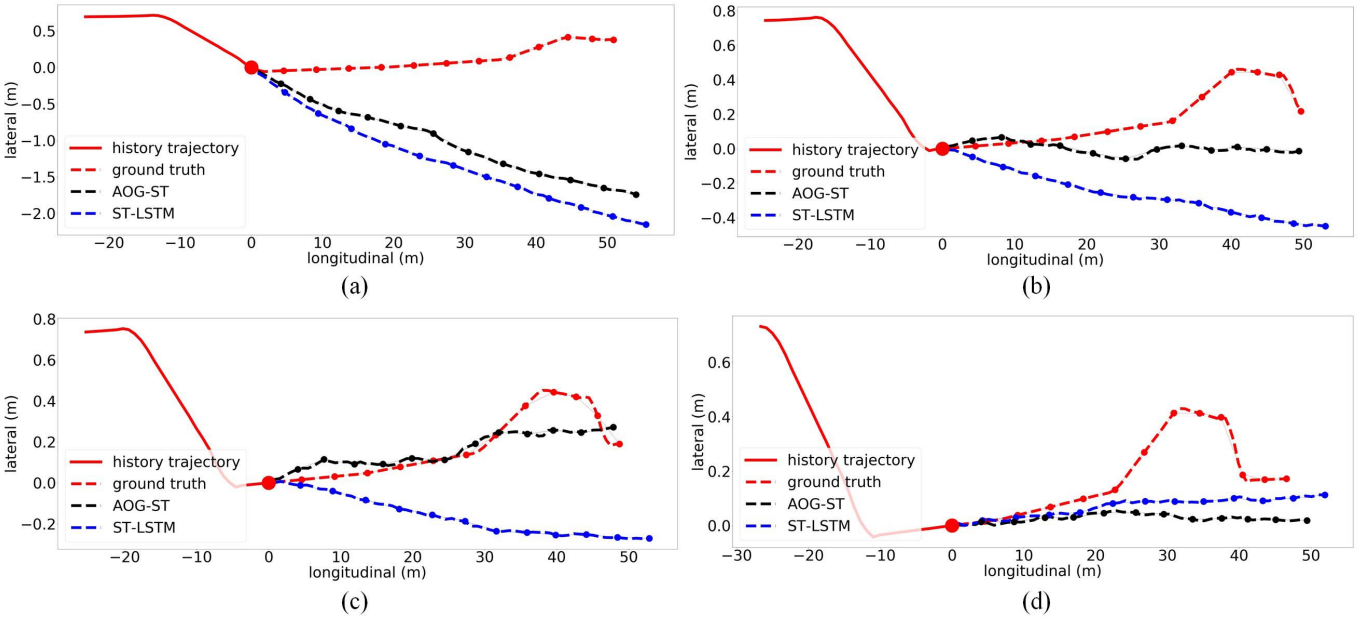



Fig. 7. The illustration of the predicted trajectories of AOG-ST and ST-LSTM when sudden change occurs: (a) $T = 0s$, (b) $T = 0.4s$, (c) $T = 0.8s$, (d) $T = 1.4s$.

$$RMS^P = \left( \frac{1}{N_{test}} \sum_{j=1}^{N_{test}} ms_j^P \right)^{\frac{1}{2}} \tag{34}$$

where $\boldsymbol{\sigma}_i^t = \left[ X^{(t+i\Delta t)} - \hat{X}^{(t+i\Delta t)}, Y^{(t+i\Delta t)} - \hat{Y}^{(t+i\Delta t)} \right]^T$. $ms^P$ defines the mean-square error of one specific trajectory in the prediction horizon $P$. $N_{test}$ is the number of trajectories in the testing set.

The RMS testing result of the three models is shown in Table II and indicates that the AOG-ST (AOG-ST-TB) has a lower prediction error than ST-LSTM.

We also calculate the prediction consuming time of the two models. Specifically, the average time of ST-LSTM to perform a single trajectory prediction with the horizon of 6$s$ is 8.16$ms$ on a TITAN X (Pascal) GPU. Note that the above value includes the consuming time of the preliminary prediction for six surrounding vehicles around the target vehicle. In practical applications, when we perform prediction in a certain area, the above preliminary trajectories can be reused. Therefore, the apportioned consuming time of a single prediction should be about half of this value (i.e., 4.08$ms$) since the calculations of correction sequences cannot be shared. This value is closely related to vehicle density, that is, it will get lower when the vehicle is more sparse.

The average consuming time of one complete maneuver inference (of our trained AOG) is 3.43 $ms$. Thus the consuming time of AOG-ST is 7.51$ms$, which is not much different from ST-LSTM.

We check some prediction instances to find the origin of the prediction error. Then we find that AOG-ST has another significant improvement which is not reflected by the numerical RMS value. Specifically, the AOG-ST is more sensitive to the sudden change of the target vehicle's driving behavior than ST-LSTM.

TABLE II
RMS VALUE OF PREDICTION ERROR

| Prediction horizon(s) | RMS value (m) | | |
|---|---|---|---|
| | ST-LSTM | AOG-ST | AOG-ST-TB |
| 1 | 0.54 | 0.53 | 0.49 |
| 2 | 1.16 | 1.14 | 1.06 |
| 3 | 1.88 | 1.83 | 1.72 |
| 4 | 2.70 | 2.61 | 2.47 |
| 5 | 3.63 | 3.51 | 3.34 |
| 6 | 4.64 | 4.48 | 4.30 |

For example, suppose the target vehicle performs a sudden cut-in to the right lane (RLC), AOG-ST can quickly notice the change by motion inference. Then AOG-ST switches to a reasonable RLC trajectory prediction model and thus responds rapidly to the new driving behavior on the predicted trajectories. In contrast, ST-LSTM cannot discover the change until the historical (hidden) information of RLC accumulates to a certain amount and starts to dominate the predicted results.

Other than significant lane changes, AOG-ST can even sensitively detect lateral jitters when the target vehicle is going straight. To demonstrate this characteristic, we randomly select a lane-following trajectory segment with obvious lateral jitters, take several snapshots in the time interval and compare the predicted trajectories by AOG-ST and ST-LSTM in Fig. 7. To highlight the difference between the prediction results of the two results, we only plot the trajectory of the target vehicle, but actually, the surrounding vehicles do exist and are considered during trajectory prediction.

As shown in Fig. 7, the target vehicle follows the specific lane in the trajectory segment, but its lateral velocity direction

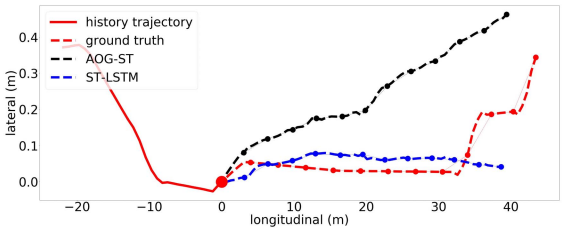



Fig. 8. The illustration of a false alarm prediction generated by AOG-ST.

is not constant. The target vehicle presents an obvious tendency to the right in Fig. 7 (a), thus both of the two models predict RLC trajectories. However, the tendency to the right does not last a long time and the lateral velocity direction change to the left, as shown in Fig. 7 (b)-(d). AOG-ST notices this change rapidly and predicts a GS trajectory instead of RLC, while the ST-LSTM still keeps predicting an RLC trajectory for about 1 second. This example demonstrates the sensitivity of AOG-ST to the sudden change of vehicle movements.

However, we find that AOG-ST is more likely to generate **"false alarm"** prediction than ST-LSTM due to the sensitivity to sudden change. As shown in Fig. 8, despite the lateral velocity direction of the target vehicle is indeed leftward at the predicting time, the lateral offset is just a jitter of the straight trajectory rather than the beginning of an LLC. However, AOG-ST captures this offset and determines that as the sign of LLC, thus outputs an LLC trajectory. The predicted trajectory has significant differences with the ground truth trajectory and may cause a large RMS error.

### C. The Discussion on False Alarm Predictions

In this subsection, to discuss the impact of "false alarm" on the overall prediction error, we do two experiments to respectively calculate the frequency of false alarm prediction and the RMS error caused by false alarm prediction of the compared models. We will demonstrate that although the AOG-ST model does generate more false alarm predictions than ST-LSTM, the false alarms do not lead to a significant increase in the overall prediction error. In fact, the RMS errors caused by false alarms in the two models are roughly equal.

(1) *The proportion of false alarm predictions*

We conduct statistics of three kinds of prediction on the testing set (**False alarm**, **Undetected** and **Others**), whose proportions are shown in Table III. Here, **"False alarm"** presents the cases that the predicted trajectory has significant lateral velocity while the ground truth movement is going straight (i.e., the cases like Fig. 8). In contrast, **"Undetected"** presents the cases that the predicted trajectory has small lateral velocity while the ground truth movement is LLC or RLC. **"Others"** include the rest types of trajectory predictions.

The AOG-ST-TB in Table III is generated based on AOG-ST by trimming some redundant clustered sub-maneuvers. The reason we trim is that these redundant sub-maneuvers increase the omission rate (the proportion of **Undetected**) of AOG-ST, as shown in Table III. The specific reason of this phenomenon is that our clustering algorithm may aggregate some abnormal GS movements (e.g. GS with lateral jitters) and generate redundant sub-maneuvers. These abnormal trajectory points are outliers relative to the general GS movements, but they may be adjacent to LLC (RLC) movements in the feature space. Therefore, some LLC (RLC) movements are inferred to these redundant GS sub-maneuvers, resulting in missed detection. Although our AOG initialization and dynamic maintenance algorithms have designed corresponding approaches to filter out the outliers by setting buffers, the algorithms cannot filter out moderate lateral fluctuations since they are actually not rare.

TABLE III
THE PROPORTION OF DIFFERENT KINDS OF PREDICTION

| Prediction types | ST-LSTM | AOG-ST | AOG-ST-TB |
|---|---|---|---|
| **False alarm** | 14.6% | 21.9% | 20.1% |
| **Undetected** | 4.2% | 5.2% | 4.0% |
| **Others** | 81.2% | 72.8% | 75.8% |

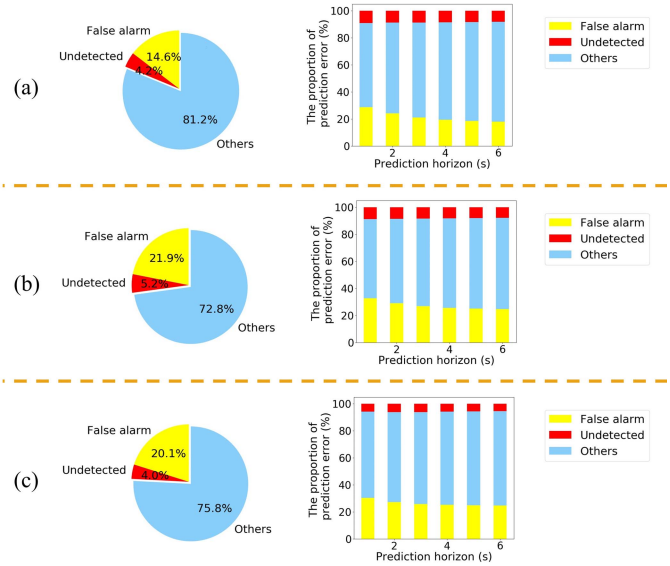



Fig. 9. The illustration of the impact of different prediction statuses (False alarm, Undetected and Others) on RMS prediction error, including the pie chart of different statuses and the bar chart of the proportion to total RMS error of different statuses: (a) ST-LSTM; (b) AOG-ST; (c) AOG-ST-TB.

Table III indicates that the AOG-ST (AOG-ST-TB) is more likely to produce false alarms than ST-LSTM, which is in line with our previous analysis. The trimmed AOG-ST-TB can maintain a similar omission rate to ST-LSTM.

(2) *The RMS error caused by false alarm predictions*

To study whether the increased false alarm predictions of AOG-ST has a significant negative impact on the prediction task, we calculate the contribution of the three kinds of predictions to the total RMS error, which is shown in Fig. 9 and Table IV-VI.

TABLE IV
RMS ERROR ANALYSIS OF ST-LSTM

| Prediction horizon (s) | $RMS^P$ (m) | **Others** $r_0^P$ (m) | | **False alarm** $r_1^P$ (m) | | **Undetected** $r_2^P$ (m) | |
|---|---|---|---|---|---|---|---|
| | | value | proportion | value | proportion | value | proportion |
| 1 | 0.54 | 0.43 | 62.2% | 0.29 | 28.8% | 0.16 | 9.0% |
| 2 | 1.16 | 0.95 | 67.1% | 0.57 | 24.1% | 0.34 | 8.7% |
| 3 | 1.88 | 1.68 | 70.3% | 0.86 | 21.0% | 0.55 | 8.6% |
| 4 | 2.70 | 2.29 | 72.0% | 1.19 | 19.5% | 0.79 | 8.5% |
| 5 | 3.63 | 3.10 | 73.1% | 1.56 | 18.5% | 1.05 | 8.3% |
| 6 | 4.64 | 3.99 | 73.8% | 1.96 | 17.9% | 1.33 | 8.2% |

TABLE V
RMS ERROR ANALYSIS OF AOG-ST

| Prediction horizon (s) | $RMS^P$ (m) | **Others** $r_0^P$ (m) | | **False alarm** $r_1^P$ (m) | | **Undetected** $r_2^P$ (m) | |
|---|---|---|---|---|---|---|---|
| | | value | proportion | value | proportion | value | proportion |
| 1 | 0.53 | 0.41 | 58.5% | 0.31 | 32.7% | 0.15 | 8.8% |
| 2 | 1.14 | 0.91 | 62.4% | 0.63 | 29.1% | 0.32 | 8.5% |
| 3 | 1.83 | 1.50 | 64.7% | 0.97 | 26.9% | 0.50 | 8.4% |
| 4 | 2.61 | 2.16 | 66.1% | 1.35 | 25.7% | 0.71 | 8.2% |
| 5 | 3.51 | 2.92 | 67.0% | 1.79 | 25.0% | 0.93 | 8.0% |
| 6 | 4.48 | 3.75 | 67.4% | 2.27 | 24.7% | 1.17 | 7.9% |

TABLE VI
RMS ERROR ANALYSIS OF AOG-ST-TB

| Prediction horizon (s) | $RMS^P$ (m) | **Others** $r_0^P$ (m) | | **False alarm** $r_1^P$ (m) | | **Undetected** $r_2^P$ (m) | |
|---|---|---|---|---|---|---|---|
| | | value | proportion | value | proportion | value | proportion |
| 1 | 0.49 | 0.39 | 63.8% | 0.27 | 30.0% | 0.12 | 5.9% |
| 2 | 1.06 | 0.87 | 66.6% | 0.56 | 27.3% | 0.26 | 6.2% |
| 3 | 1.72 | 1.41 | 68.0% | 0.87 | 25.9% | 0.42 | 6.1% |
| 4 | 2.47 | 2.05 | 68.9% | 1.24 | 25.3% | 0,60 | 5.9% |
| 5 | 3.34 | 2.79 | 69.4% | 1.67 | 24.9% | 0.80 | 5.7% |
| 6 | 4.30 | 3.59 | 69.8% | 2.14 | 24.8% | 1.01 | 5.5% |

In Table IV-VI, the RMS-like statistics $r_0^P$, $r_1^P$ and $r_2^P$ are used to measure the specific value of the RMS error caused by the three kinds of predictions:

$$ms^P = \frac{1}{P}\sum_{i=1}^{P}\left(\boldsymbol{\sigma}_i^t\right)^T \boldsymbol{\sigma}_i^t \tag{35}$$

$$r_k^P = \left(\frac{1}{N_{test}} 1\left\{l_j = k\right\} \sum_{j=1}^{N_{test}} ms_j^P\right)^{\frac{1}{2}} \tag{36}$$

where $k \in \{0, 1, 2\}$. $l_j$ is the prediction status (i.e., one of **False alarm**, **Undetected** and **Others**) given the historical trajectory $j$. It is straightforward to get the relationship between $RMS^P$ and $r_k^P$:

$$\left(RMS^P\right)^2 = \sum_k \left(r_k^P\right)^2 \tag{37}$$

Therefore, the proportion (denoted as $p_k$) of error caused by prediction status $k$ to the total RMS error is calculated by:

$$p_k = \left(r_k^P / RMS^P\right)^2 \tag{38}$$

The tables and figures demonstrate that although the false alarm prediction of AOG-ST (AOG-ST-TB) accounts for a larger proportion than ST-LSTM, the RMS error value caused by false alarms in the three models are almost equal. In contrast, the RMS errors generated by undetected and other predictions have significant reductions in AOG-ST, which demonstrates AOG-ST yields better performance than ST-LSTM.

We examine a large number of trajectory instances that AOG-ST (AOG-ST-TB) and ST-LSTM present different kinds of prediction. Most of these trajectories have significant lateral oscillation and be different from both typical GS and LLC (RLC), such as the case in Fig. 8. When performing trajectory predictions, AOG-ST is aggressive to determine these motions as LLC (RLC) and generate corresponding trajectories. In contrast, since ST-LSTM cannot actively infer the motions, it keeps generating GS predictions until accumulating enough lateral fluctuation information. Both of these two prediction results may be dissimilar with the ground truth trajectory and produce large but comparable prediction errors. However, from the perspective of forewarning, this sensitive false alarm is beneficial, which helps to improve the security of decision making and planning.

### D. The Compatibility of AOG-ST With Other Scenarios

In this paper, we only train and evaluate our AOG-ST on a "six surrounding vehicle scenario". However, it does not mean that AOG-ST cannot handle scenarios with other numbers of surrounding vehicles or other types of roads, nor does it mean that a trained AOG-ST cannot normally run when less than six surrounding vehicles are available. In fact, AOG-ST models (whether have been trained or not) have great flexibilities. We will respectively explain the flexibility of each part of AOG-ST as follows:

(1) *AOG*

AOG only defined the semi-supervised learning of sub-maneuvers and the hierarchical inference method of multi-level labels. Since the vehicle spatial interactions are extracted as input features of AOG, there is no direct relation between the number of surrounding vehicles and the structure and calculation of AOG.

(2) *The interaction feature extraction method for AOG*

The interaction feature (prior lane change probabilities) extraction algorithm models the impact of all surrounding vehicles as potential functions, and then performs a lane-wise superimposition to represent the objective driving condition of each lane. Therefore, it is straightforward that the number of surrounding vehicles is not limited since the superposition method allows any number of potential values.

If we want to model other types of roads, we should change the interaction features according to the prior knowledge of vehicle movements and some AOG hyperparameters rather than modifying the AOG structure.

(3) *ST-LSTM*

ST-LSTM serializes and models the spatial impact of each surrounding vehicle as a trajectory correction sequence, which semantically means the potential correction of the target vehicle to avoid the safety risk caused by this surrounding vehicle. The calculation of the correction sequences are independent, and they are superimposed by a set of normalized dynamic weights. Therefore, the prediction can adapt to any number of surrounding vehicles.

Therefore, our overall model can theoretically adapt to any scenario after appropriate modifications of hyper-parameters or features. However, as we introduced in [4], ST-LSTM has a critical training set volume to avoid severe overfitting, while the data volume requirement of AOG-ST is much larger than that of a single ST-LSTM. We have not found an open-source dataset in other scenarios that have similar volume and quality to NGSIM I-80 (and US-101). We have trained our model on some (self-collected or simulated) small-scale dataset in other scenarios, but they are proven not enough to well train AOG-ST. Therefore, we only display the results of straight road scenarios in this paper.

## VII. Concluding Remarks

In this paper, we propose a new flexible and explainable framework for motion prediction. We inherit the merit of using deep leering from end-to-end approaches to build flexible and accurate prediction models and meanwhile formulate explainable maneuver divisions for prediction results. We address four difficulties of maneuver-based models and propose four corresponding solutions.

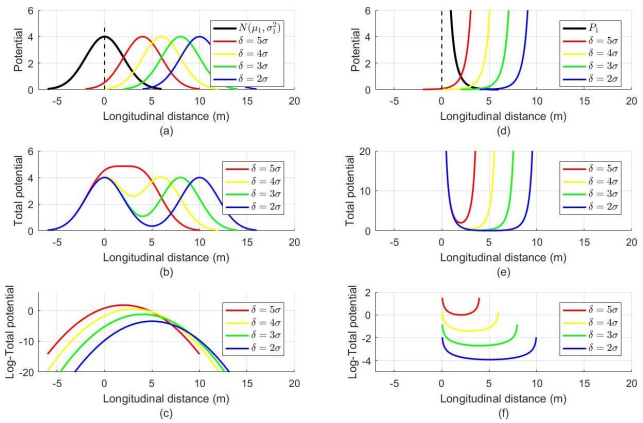



Fig. 10. The illustration of 1-D potential superposition, $\mu_1 = 0$, $\sigma_1 = \sigma_2 = 2$, $\delta = \mu_2 - \mu_1$: (a)-(c): Gaussian-like functions, (d)-(f): Inverse proportional-like functions, (a)(d): The original potential functions, (b)(e): The combined potential by additive superposition, (c)(f): The combined potential by multiplicative superposition (in logarithmic coordinates).

However, our work also has some limitations:

First, we only test our model in the scenario of straight roads, due to the limited testing data. However, we have shown that AOG-ST does not restrict itself to handle scenarios with a certain number of surrounding vehicles or a certain type of environment information in *Section IV-D*. In our future work, we will collect data from different scenarios and perform a more comprehensive evaluation of AOG-ST.

Second, our trajectory predictions do not consider road information [38] due to the limitation of the data set. Moreover, it is hard to encode and utilize such information for current motion prediction models. In fact, road information is very important in the prediction of complex scenarios. In our future work, we will also explore the methods to utilize road information based on our deeper understanding of the influence mechanism of different road information on human driving.

In addition, we believe that the quantitative evaluation of false alarm predictions is also an issue worth studying. We have qualitatively discussed in *Section IV-C* that for critical trajectory cases that contain significant driving oscillations, the boundary between false alarm predictions and timely forewarnings is blurred. However, whether it is actually a false alarm or a successful forewarning, an appropriate sensitivity of aggressively discovering the potential behavior changes from trajectories is worth encouraging (while excessive sensitivity is not recommended). In future works, we will further investigate the impact of this sensitivity and try to find a quantitative index to guide the corresponding model design and evaluation.

## Appendix A
## The Comparison of Two Types of Vehicle Potential Functions

In conventional works, there are two types of commonly used vehicle potential functions: inverse proportional-like

**Algorithm 1** AOG Initialization

**Input**: $F$, $N_{sub}$.
$F$: The training set with scenario and maneuver labels.
$N_{sub}$: The maximum number of sub-maneuvers subordinate to one maneuver.
**Output**: AOG includes $\mathcal{M}$, $\mathcal{H}_{j,k}$, $\mathcal{G}_{j,k}$, $STG(\mathcal{M})$ $STG(\mathcal{H}_{j,k})$.
$\mathcal{M} = \{M_k | 1 \le k \le n_k\}$: maneuver set (contains one unclassified node to store unlabeled data).
$\mathcal{H}_{j,k} = \left\{H^i_{j,k} | 1 \le j \le n_{j,k}\right\}$: sub-maneuver set subordinate to $M_k$ (contains one unclassified node).
$\mathcal{G}_{j,k}$: Gaussian distributions set corresponding to $\mathcal{H}_{j,k}$.
$STG(\mathcal{M})$: The state transition graph of $STG(\mathcal{M})$.
$STG(\mathcal{H}_{j,k})$: The state transition graph of $\mathcal{H}_{j,k}$.

Construct $\mathcal{M}$ according to the known maneuver labels;
**for** $f$ *in* $F$ **do**
  Store $f$ into $\mathcal{M}$ according to its maneuver label;
**end**
**for** $k = 1:1:n_k$ **do**
  $i := 0$;
  Perform DBSCAN (*Section IV-A*) on $M_k$ and obtain sub-maneuvers $\mathcal{H}^i = \left\{H^i_{j,k} | 1 \le j \le n_{j,k}\right\}$ after filtering out the noise classes;
  Fit Gaussian distributions $\mathcal{G}^i$ for $\mathcal{H}^i$;
  **while** $n_{j,k} < N_{sub}$ **do**
    $i := i + 1$;
    **for** $h$ *in* $\mathcal{H}^{i-1}$ **do**
      Perform node splitting (*Section IV-A*) on $h$;
    **end**
    Obtain new sub-maneuvers $\mathcal{H}^i$ (with $\mathcal{G}^i$);
    **if** $\mathcal{H}^i = \mathcal{H}^{i-1}$ **then**
      break;
    **end**
  **end**
  $\mathcal{H}_{j,k} := \mathcal{H}^i$ and $\mathcal{G}_{j,k} := \mathcal{G}^i$;
  Generate $STG(\mathcal{H}_{j,k})$ (use **Algorithm 2**);
**end**
Generate $STG(\mathcal{M})$ (use **Algorithm 2**);
return the AOG.

**Algorithm 2** STG Initialization

**Input**: $\mathcal{N}$, $F_{\mathcal{N}}$, $N_{style}$
$\mathcal{N} = \{N_k | 1 \le k \le n_{\mathcal{N}}\}$: The set of patterns with state transition relationships.
$F_{\mathcal{N}}$: The training set of this STG.
$N_{style}$: The number of state transition styles.
**Output**: $STG(\mathcal{N})$ includes $\mathcal{P}$, $\mathcal{S}$.
$\mathcal{P} = \left\{(N_i, N_j) | N_i, N_j \in \mathcal{N}\right\}$: The set of state transition pairs within $\mathcal{N}$.
$\mathcal{S} = \left\{f_r(N_i, N_j, d_i) | N_i, N_j \in \mathcal{N}, 1 \le r \le N_{style}\right\}$: The set of state transition probability functions (values) corresponding to $\mathcal{P}$.
$f_r(N_i, N_j, d_i)$: The transition probability function (value) from $N_i$ to $N_j$ given transition style $r$ and duration $d_i$.

**for** $f$ *in* $F_{\mathcal{N}}$ **do**
  Calculate the observation probability of $f$ by AOG and obtain the most-likely pattern $N_{now}$;
  **if** *a state transition is captured* **then**
    Record the state transition pair $(N_{old}, N_{now})$ and the duration of $N_{old}$ in $\mathcal{P}$;
  **end**
**end**
**for** $P = (N_i, N_j)$ *in* $\mathcal{P}$ **do**
  Obtain the duration sequence $S(N_i, N_j)$ of $P$;
  **if** $N_i$ *is a non-trivial node* **then**
    Cluster $S(N_i, N_j)$ into $N_{style}$ classes;
    **for** $r = 1:1:N_{style}$ **do**
      Fit $f_r(N_i, N_j, d_i)$ based on LR (*Section III-C*);
    **end**
  **end**
  **else**
    Constant value $f_r(N_i, N_j, d_i) \propto \frac{\text{length}(S(N_i, N_j))}{\sum_{n \in succ(N_i)} \text{length}(S(N_i, n))}$
  **end**
**end**
Constant value $f_r(n, n, d_i) = \frac{T(n)}{T}$ where $n$ is the trivial node in STG, $T(n)$ is the total time of pattern $n$ and $T$ is the total time of $F_{\mathcal{N}}$;
Normalization of state transition probabilities of the trivial node;
return the $STG(\mathcal{N})$.

functions and Gaussian-like functions. The main difference between the two types of functions is the values of peaks.

**Inverse proportional-like functions:** Suppose $r$ is one distance measure between $V_i$ and $V_s$. Rasekhipour *et al.* [39] use the function of $\frac{A}{r}$ and Wolf *et al.* adopt $\frac{A\exp(-r)}{r}$ to construct the potential field generated by $V_i$. Wang *et al.* also use a similar form [18], [40]. The potential values of this kind of functions approach to infinity as $r$ decreases, which indicates the center of the potential field, i.e., the position of $V_i$ is uncrossable.

**Gaussian-like functions:** Wahid *et al.* [41] use the probability density function of 2-D Gaussian distribution to construct the potential field. This kind of functions have a finite peak, thus Rasekhipour *et al.* [39] also use them to model crossable obstacles.

The influence of peak value appears when performing potential superposition. Take 1-D Gaussian potential functions as an example, let $N\left(\mu_1, \sigma_1^2\right)$ and $N\left(\mu_2, \sigma_2^2\right)$ be the potential field generated by $V_1$ and $V_2$. Additive superposition and multiplicative superposition are two commonly used potential superposition methods in conventional works. The corresponding combined Gaussian potential functions are:

$$f_+(x) = a_1 e^{-\frac{(x-\mu_1)^2}{\sigma_1^2}} \left(1 + \frac{a_2}{a_1} e^{-\left(\frac{\sigma_1}{\sigma_2}\right)^2 \frac{(x-\mu_1)^2}{(x-\mu_2)^2}}\right) \quad (39)$$

$$f_{\mathbf{x}}(x) = a_1 a_2 e^{-\frac{(x-\mu_1)^2}{\sigma_1^2} - \frac{(x-\mu_2)^2}{\sigma_2^2}} \quad (40)$$

**Algorithm 3** AOG Dynamic Maintenance

**Input**: $F_{new}$, $N_{buf}$
$F_{new}$: The new training set to update the AOG.
$N_{buf}$: The maximum volume of each AOG node to store new data.
**Output**: The updated AOG.

```
for f in F_new do
    if f has a new maneuver label then
        Store f;
        if Stored f more than N_buf then
            Construct the sub-maneuvers of the new
            maneuver in AOG based on Algorithm 1
        end
    end
    else
        Inference the maneuver m_f and the sub-maneuver
        h_f based on AOG (Section III-B);
        Store f in h_f;
        if Stored f more than N_buf then
            Update the Gaussian distribution of h_f ;
            Update functions f_r(h_f, succ(h_f), d_i) and
            f_r(pred(h_f), h_f, d_i) of the STG;
            for h' ≠ h_f subordinate to m_f do
                h_f := merge(h_f, h') if the merging success
                (Section IV-A); update the STG;
            end
            Perform node splitting (Section IV-A) on h_f;
            update the STG;
        end
    end
end
return the updated AOG.
```

As shown in Fig. 10 (a)-(c), when $V_2$ approaches to $V_1$ (i.e, $\mu_2 \to \mu_1$), both $f_+(x)$ and $f_\times(x)$ approach to Gaussian functions and a new peak appears between $\mu_1$ and $\mu_2$. This means that the region between $V_1$ and $V_2$ is even more dangerous than the region where $V_1$ or $V_2$ is located, which is obviously contrary to our intuition. In contrast, as shown in Fig. 10 (d)-(f), inverse proportional-like functions do not have this problem. Therefore, the inverse proportional-like functions are more suitable for the scenarios that have large amounts of potential superposition. Especially, we adopt the function $\frac{A\exp(-r)}{r}$ in this paper.

## APPENDIX B
## THE NOMENCLATURE LIST

| Symbol | Definition |
|---|---|
| **The basic notations of the trajectory prediction problem** | |
| $V_s$ | The target vehicle of trajectory prediction. |
| $V_i$ | One surrounding vehicle of $V_s$ ($i \in \{fl, rl, fs, rs, fr, rr\}$). |
| $\left[X_s^t, Y_s^t\right]$ ($\left[X_i^t, Y_i^t\right]$) | The ground truth coordinates of $V_s$ ($V_i$) at the time $t$. |
| $v_{X,s}^t, v_{Y,s}^t$ ($v_{X,i}^t, v_{Y,i}^t$) | The ground truth lateral and longitudinal velocities of $V_s$ ($V_i$) at the time $t$. |
| $\Delta t$ | The time resolution level. |
| $\left[\hat{X}_s^{(t+\Delta t)}, \hat{Y}_s^{(t+\Delta t)}\right]$ | The predicted coordinates of $V_s$ at the time $(t+\Delta t)$. |
| $t_h$ | The length of historical time horizon. |
| $t_p$ | The prediction time horizon. |
| $\boldsymbol{I}^t$ | The input of the trajectory prediction model (a $\frac{t_h+\Delta t}{\Delta t} \times 28$ matrix). |
| $\boldsymbol{f}^t$ | The extracted feature vector for AOG (a $8 \times 1$ vector). |
| $\hat{\boldsymbol{o}}_s^t$ | The output of the trajectory prediction model (a $2t_p \times 1$ vector). |
| $m^t$ ($h^t$) | The estimated maneuver (sub-maneuver) of $V_s$ at the time $t$. |
| $\boldsymbol{o}_s^t$ | The ground truth future trajectory of $V_s$. |
| $M^t$ | The manually labeled maneuver of $V_s$ at the time $t$. |
| $\boldsymbol{x}_s^t$ ($\boldsymbol{x}_i^t$) | The input of ST-LSTM trajectory prediction model (derived from $I^t$). |
| $\boldsymbol{y}_s^t$ | The output of ST-LSTM trajectory prediction model (calculate $\hat{\boldsymbol{o}}_s^t$). |
| **The basic notations of AOG** | |
| $S$ | The research scenario. |
| $\{M_k \mid 1 \le k \le n_k\}$ | The maneuvers of AOG (learned from the maneuver labels of the training set). |
| $\{H_{j,k} \mid 1 \le j \le n_{j,k}\}$ | The sub-maneuvers that subordinate to maneuver $M_k$ (learned by then semi-supervised algorithm). |
| $N_{style}$ | The number of And-nodes (configurations) under each Or-nodes (maneuvers and sub-maneuvers) in AOG. |
| $Z_r$ | One configuration (style) that belongs to $S$ ($1 \le r \le N_{\text{style}}$). |
| $Z_{r,k}$ | One configuration (style) that belongs to $M_k$ ($1 \le r \le N_{\text{style}}$). |
| **The parameters of AOG inference** | |
| $P\left(\boldsymbol{f}^t \mid h_k^t = H_{i,k}\right)$ | The emission probability of sub-maneuver $H_{j,k}$. |
| $P\left(h_k^t = H_{j,k} \mid \boldsymbol{f}^t\right)$ | The posterior probability of $H_{j,k}$ given $\boldsymbol{f}^t$ (also denoted as $P_{j,k}^t$). |
| $d_{j,k}$ | The duration of $H_{j,k}$ up to the time $(t-\Delta t)$. |
| $P\left(z_k^{t-\Delta t} = Z_{r,k}\right)$ | The probability that $V_s$ adopts the style $Z_{r,k}$ up to the time $(t-\Delta t)$. |
| $P\left(H_{j,k} \to H_{i,k}\right)$ | The transition probability from $H_{j,k}$ to $H_{i,k}$ regardless the state transition style of $H_{i,k}$. |
| $P_r\left(H_{j,k} \to H_{i,k}\right)$ | The transition probability from $H_{j,k}$ to $H_{i,k}$ under the state transition style $Z_{r,k}$. |

| | |
|---|---|
| $\hat{h}_k^t$ | The most-likely sub-maneuver among $H_{j,k}$ (below $M_k$). |
| $\boldsymbol{h}^t$ | The candidate sub-maneuver vector that stores the most likely sub-maneuver $\hat{h}_k^t$ below each $M_k$. |
| $P\left(\boldsymbol{h}^t \mid m^t = M_k\right)$ | The emission probability of maneuver $M_k$. |
| $P\left(m^t = M_k \mid \boldsymbol{h}^t\right)$ | The posterior probability of $M_k$ (also denoted as $P_k^t$). |
| $d_k$ | The duration of $M_k$ up to the time $(t - \Delta t)$. |
| $P\left(z^{t-\Delta t} = Z_r\right)$ | The probability that $V_s$ adopts the style $Z_r$ up to the time $(t - \Delta t)$. |
| $P\left(M_k \rightarrow M_i\right)$ | The transition probability from $M_k$ to $M_i$ regardless the state transition style of $M_k$. |
| $P_r\left(M_k \rightarrow M_i\right)$ | The transition probability from $M_k$ to $M_i$ under the state transition style $Z_r$. |
| $\hat{m}$ | The most-likely maneuver among $M_k$. |
| $\alpha_{h\rightarrow h'}, \beta_{h\rightarrow h'}$ | The parameters of the state transition probability function that corresponding to the transition $h \rightarrow h'$. |
| $\mathcal{L}$ | The loss function to optimize $\alpha_{h\rightarrow h'}$ and $\beta_{h\rightarrow h'}$. |
| $N_d$ | The number of times that the transition $h \rightarrow h'$ occurs in the training set. |
| $\{T_i \mid 1 \leq i \leq N_d\}$ | The generated dataset to optimize $\alpha_{h\rightarrow h'}$ and $\beta_{h\rightarrow h'}$. |
| **The parameters of AOG initialization and dynamic maintenance** | |
| $\varepsilon$ | The distance threshold of DBSCAN for AOG initialization. |
| $n$ | The minimum number of points in each cluster generated by DBSCAN. |
| $G$, $G'$ | Two candidate models to be compared by BIC. $G$ is a single Gaussian model and $G'$ is a GMM that mixed by $G_1$ and $G_2$. |
| $\theta$, $\theta_1$, $\theta_2$ | The parameters of $G$, $G_1$ and $G_2$. |
| $Th_s$ | The threshold of AOG node splitting. |
| $Th_m$ | The threshold of AOG node merging. |
| $N_{sub}$ | The maximum number of sub-maneuvers subordinate to one maneuver. |
| $N_{buf}$ | The maximum volume of each AOG node to store new unclassified data. |
| **The parameters of feature extraction** | |
| $D_x$, $D_y$ | The lateral and longitudinal safe distance between two adjacent vehicle $V_s$ and $V_i$. |
| $a_{brake}$ | The maximum deceleration rate of the follower vehicle and the leader vehicle. |
| $\rho$ | The response time of the following vehicle. |
| $L_{lane}$ | The width of the lane. |
| $A$ | The coefficient of the potential functions. |
| $\Theta_i$ | The potential field generated from $V_i$. |
| $\vartheta_i$ | The potential value of $V_s$ generated from $\Theta_i$. |
| $\alpha_{l1}, \alpha_{l2}, \alpha_{r1}, \alpha_{r2}$, $\beta_s, \beta_l, \beta_r$ | The parameters to calculate the prior left (right) lane change probabilities. |
| $P_{rlc}$ $(P_{llc})$ | The prior probability of right (left) lane change considering surrounding vehicles. |

## APPENDIX C
## THE ALGORITHMS

**Algorithm 1:** *The AOG initialization algorithm*
**Algorithm 2:** *The STG initialization algorithm*
**Algorithm 3:** *The AOG dynamic maintenance algorithm*

## REFERENCES


[1] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, p. 1, Dec. 2014.

[2] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[3] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826–1848, May 2020.

[4] 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.

[5] 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. Vehicles*, vol. 3, no. 2, pp. 129–140, Jun. 2018.

[6] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2011.

[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. 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.

[9] F. Altche and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 353–359.

[10] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 399–404.

[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] J.-H. Kim and D.-S. Kum, "Threat prediction algorithm based on local path candidates and surrounding vehicle trajectory predictions for automated driving vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 1220–1225.

[13] J. Zhang and B. Roessler, "Situation analysis and adaptive risk assessment for intersection safety systems in advanced assisted driving," in *Autonome Mobile Systeme 2009*. Berlin, Germany: Springer, 2009, pp. 249–258.

[14] 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.*, Sep. 2010, pp. 1625–1631.

[15] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "A Bayesian approach for driving behavior inference," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 595–600.

[16] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Estimation of multivehicle dynamics by considering contextual information," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 855–870, Aug. 2012.

[17] M. T. Wolf and J. W. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 3731–3736.

[18] J. Wang, J. Wu, and Y. Li, "The driving safety field based on driver–vehicle–road interactions," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2203–2214, Aug. 2015.

[19] J. Wang, J. Wu, X. Zheng, D. Ni, and K. Li, "Driving safety field theory modeling and its application in pre-collision warning system," *Transp. Res. Part C, Emerg. Technol.*, vol. 72, pp. 306–324, Nov. 2016.

[20] 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, Jun. 1997, p. 994.

[21] N. Oliver and A. P. Pentland, "Graphical models for driver behavior recognition in a SmartCar," in *Proc. IEEE Intell. Vehicles Symp.*, Oct. 2000, pp. 7–12.

[22] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 1162–1167.

[23] 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: http://arxiv.org/abs/1801.04340

[24] S. Wang, L. Li, N.-N. Zheng, and D. Cao, "Modeling and predicting vehicle motion activities by using and-or graph," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 2125–2130.

[25] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Found. Trends Comput. Graph. Vis.*, vol. 2, no. 4, pp. 259–362, 2006.

[26] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 797–802.

[27] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Driver behavior classification at intersections and validation on large naturalistic data set," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 724–736, Jun. 2012.

[28] 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.

[29] C. Liu and M. Tomizuka, "Enabling safe freeway driving for automated vehicles," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 3461–3467.

[30] J. Gan and Y. Tao, "Dynamic density based clustering," in *Proc. ACM Int. Conf. Manage. Data (SIGMOD)*, 2017, pp. 1493–1507.

[31] E. Lughofer, "Dynamic evolving cluster models using on-line split-and-merge operations," in *Proc. 10th Int. Conf. Mach. Learn. Appl. Workshops*, vol. 2, Dec. 2011, pp. 20–26.

[32] A. Doshi and M. M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 453–462, 2009.

[33] J. D. Lee *et al.*, *Distraction Detection and Mitigation Through Driver Feedback*, 2013.

[34] Y. Zheng, A. Sathyanarayana, and J. Hansen, "Non-uniform time window processing of in-vehicle signals for maneuvers recognition and route recovery," SAE Tech. Paper 2015-01-0281, 2015.

[35] M. Hu, G. Xie, H. Gao, D. Cao, and K. Li, "Manoeuvre prediction and planning for automated and connected vehicles based on interaction and gaming awareness under uncertainty," *IET Intell. Transp. Syst.*, vol. 13, no. 6, pp. 933–941, Jun. 2019.

[36] J. Zhang and B. Roessler, "Situation analysis and adaptive risk assessment for intersection safety systems in advanced assisted driving," in *Autonome Mobile Systeme*. Springer, 2009, pp. 249–258.

[37] L. Li, X. Peng, F.-Y. Wang, D. Cao, and L. Li, "A situation-aware collision avoidance strategy for car-following," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 5, pp. 1012–1016, Sep. 2018.

[38] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *Proc. Conf. Robot Learn.*, 2018, pp. 947–956.

[39] Y. Rasekhipour, A. Khajepour, S.-K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1255–1267, May 2017.

[40] J. Wang, J. Wu, Y. Li, and K. Li, "The concept and modeling of driving safety field based on driver-vehicle-road interactions," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 974–981.

[41] N. Wahid, H. Zamzuri, M. A. Abdul Rahman, S. Kuroda, and P. Raksincharoensak, "Study on potential field based motion planning and control for automated vehicle collision avoidance systems," in *Proc. IEEE Int. Conf. Mechatronics (ICM)*, Feb. 2017, pp. 208–213.

[42] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[43] *Traffic Analysis Tools: Next Generation Simulation-FHWA Operations*. Accessed: Aug. 14, 2019. [Online]. Available: https://ops:fhwa:dot:gov/trafficanalysistools/ngsim:htm

**Shengzhe Dai** is currently pursuing the M.Eng. degree with the Department of Automation, Tsinghua University, Beijing, China. His research interests include machine learning, intelligent transportation systems, and intelligent vehicle.

**Zhiheng Li** (Member, IEEE) received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2009.

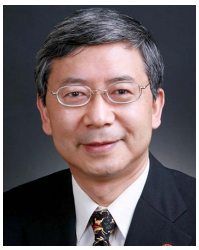
He is currently an Associate Professor with the Department of Automation, Tsinghua University, and with the Graduate School at Shenzhen, Tsinghua University, Shenzhen, China. His research interests include traffic operation, advanced traffic management systems, urban traffic planning, and intelligent transportation systems.

Dr. Li serves as an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

**Li Li** (Fellow, IEEE) received the Ph.D. degree in system and industrial engineering from The University of Arizona, Tucson, AZ, USA, in 2005.

He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China, researching fields of artificial intelligence, intelligent control and sensing, intelligent transportation systems, and intelligent vehicles. He has authored over 80 SCI indexed international journal articles and more than 50 international conference papers as a first/corresponding author.

Dr. Li serves as a member of the Editorial Advisory Board for *Transportation Research Part C: Emerging Technologies* and *Acta Automatica Sinica* and an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.



**Nanning Zheng** (Fellow, IEEE) graduated from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China, in 1975. He received the M.S. degree in information and control engineering from Xi'an Jiaotong University in 1981 and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985.

He is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His current research interests include computer vision, pattern recognition, image processing, and hardware implementation of intelligent systems.

Dr. Zheng became a member of the Chinese Academy of Engineering in 1999. He is the Chinese Representative on the Governing Board of the International Association for Pattern Recognition. He serves as an Executive Deputy Editor for the *Chinese Science Bulletin*.

**Shuofeng Wang** is currently pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing China. His research interests include intelligent transportation systems and travel behavior analysis.