2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Probabilistic Prediction of Vehicle Semantic Intention and Motion

Yeping Hu, Wei Zhan and Masayoshi Tomizuka

***Abstract*— Accurately predicting the possible behaviors of traffic participants is an essential capability for future autonomous vehicles. The majority of current researches fix the number of driving intentions by considering only a specific scenario. However, distinct driving environments usually contain various possible driving maneuvers. Therefore, a intention prediction method that can adapt to different traffic scenarios is needed. To further improve the overall vehicle prediction performance, motion information is usually incorporated with classified intentions. As suggested in some literature, the methods that directly predict possible goal locations can achieve better performance for long-term motion prediction than other approaches due to their automatic incorporation of environment constraints. Moreover, by obtaining the temporal information of the predicted destinations, the optimal trajectories for predicted vehicles as well as the desirable path for ego autonomous vehicle could be easily generated. In this paper, we propose a Semantic-based Intention and Motion Prediction (SIMP) method, which can be adapted to any driving scenarios by using semantic-defined vehicle behaviors. It utilizes a probabilistic framework based on deep neural network to estimate the intentions, final locations, and the corresponding time information for surrounding vehicles. An exemplar real-world scenario was used to implement and examine the proposed method.**

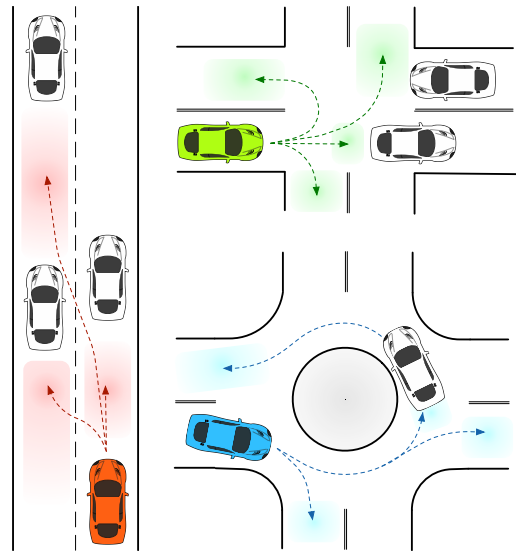


Fig. 1. Insertion areas (colored regions) under different driving scenarios for the predicted vehicle.

I. Introduction

Safety is the most fundamental aspect to consider for both human drivers and autonomous vehicles. Human drivers are capable of using past experience and intuitions to avoid potential accidents by predicting the behaviors of other drivers. However, some drivers have poor driving habits such as changing lanes without using turn signals, which adds difficulties for prediction. Moreover, human drivers might easily overlook dangerous situations due to limited concentration. Therefore, the Advanced Driver Assistance Systems (ADAS) should have the ability to simultaneously and accurately anticipate future behaviors of multiple traffic participants under various driving scenarios, which may then assure a safe, comfortable and cooperative driving experience.

There have been numerous works focused on predicting vehicle behavior which can be divided into two categories: **intention/maneuver prediction** and **motion prediction**. Many intention estimation problems have been solved by using classification strategies, such as Support Vector Machine (SVM) [1], Bayesian classifier [2], Multilayer Perceptron (MLP) [3], and Hidden Markov Models (HMMs) [4]. Most of these approaches were only designed for one particular scenario associated with limited intentions. For example, [1]-[3] dealt with non-junction segment such as highway, which involves lane keeping (LK), lane change left (LCL) and lane change right (LCR) maneuvers. Whereas [4]-[6] concentrated on junction segment such as intersection, which includes left turn, right turn, and go straight maneuvers. However, in order for autonomous vehicles to drive through dynamically changing traffic scenes in real life, an intention prediction module that can adapt to different scenarios with various possible driving maneuvers is necessary. [7] proposed a maneuver estimation approach for generic traffic scenarios, but the classified driving maneuvers are too specific, which will not only require multiple manually-selected classification thresholds, but also raise problems when unclassified maneuvers occur.

As a result, we proposed to use semantics to represent the driver intention, which is defined as the intent to enter each *insertion area*. These areas can be the available gaps between any two vehicles on the road or can be the lane entrances/exits. Fig. 1 visualizes the insertion areas under distinct environments. An advantage of using semantic approach is that situations can be modeled in a unified way [8] such that varying driving scenarios will have no effect on our semantics defined problem. Even for a scenario that has a combination of all the road structures in Fig. 1, the proposed semantic definition still holds.

Motion prediction is mostly treated as a regression problem, where it tries to forecast the short-term movements and long-term trajectories of vehicles. By incorporating motion

*This work was supported by Berkeley DeepDrive (BDD).

Y. Hu, W. Zhan and M. Tomizuka are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA [yeping_hu, wzhan, tomizuka@berkeley.edu]

prediction with intention estimation, not only the high-level behavioral information, but also the future state of the predicted vehicle can be obtained. For short-term motion prediction, various approaches such as constant acceleration (CA), Intelligent Driver Model (IDM) [6], and Particle Filter (PF) [9] have been suggested. The main limitation of these works, however, is that they either considered simple cases such as car following or did not take environment information into account.

For future trajectory estimation, Dynamic Bayesian Networks (DBN) [10] and other regression models [11] have been used in several studies. Methods based on artificial neural network (ANN) are also widely applied. In [12], the authors used the LSTM to predict the vehicle trajectory in highway situation. [13] brought forward a Deep Neural Networks (DNN) to obtain the lateral acceleration and longitudinal velocity. However, these approaches only predicted the most likely trajectory for the vehicle without considering uncertainties in the environment. To counter this issue, a Variational Gaussian Mixture Model (VGMM) was proposed for probabilistic long-term motion prediction [14]. Nevertheless, the method was only tested in a simulation environment and the input contains history information over a long period of time, which is usually unaccessible in reality. There are also researches that project the prediction step of a tracking filter forward over time, but the growing uncertainties often cause future positions to end up at some physically impossible locations.

In contrast, works such as [15] and [16] highlighted that by predicting goal locations and assuming that agents navigate toward those locations by following some optimal paths, the accuracy of long-term prediction can be improved. The main advantage of postulating destinations instead of trajectories is that it allows one to represent various dynamics and to automatically incorporate environment constraints for unreachable regions.

Apart from obtaining the possible goals of predicted vehicles, the required time to reach those locations is also an essential information especially for the subsequent trajectory planning of the ego vehicle. Therefore, many attempts have been made in order to directly predict temporal information. [17] used LSTM to forecase time-to-lane-change (TTLC) of vehicles under highway scenarios. A recent work [18] utilized the Linear Quantile Regression (LQR) and Quantile Regression Forests (QRF) methods for the probabilistic regression task of TTLC. The authors also concluded that QRF has better performance than LQR.

In this paper, Semantic-based Intention and Motion Prediction (SIMP) method is proposed. It utilizes deep neural network to formulate a probabilistic framework which can predict the possible semantic intention and motion of the selected vehicle under various driving scenarios. The introduced semantics for this prediction problem is defined as answering the question of *"Which area will the predicted vehicle most likely insert into? Where and when?"*, which incorporates both the goal position and the time information into each insertion area. Moreover, the adoption of probability can take into account the uncertainty of drivers as well as the evolution of the traffic situations.

The remainder of the paper is organized as follows: Section II provides the concept of the proposed SIMP method; Section III discusses an exemplar scenario to apply SIMP; evaluations and results are provided in Section IV; and Section V concludes the paper.

II. Concept of Semantic-based Intention and Motion Prediction (SIMP)

In this section, we first provide a brief overview of Mixture Density Network (MDN), which is an idea we utilize for our proposed method. Then, the detailed formulation and structure of the SIMP method are illustrated.

A. Mixture Density Network (MDN)

Mixture Density Network is a combination of ANN and mixture density model, which was first introduced by Bishop [19]. The mixture density model can be used to estimate the underlying distribution of data, typically by assuming that each data point has some probability under a certain type of distribution. By using a mixture model, more flexibility can be given to completely model the general conditional density function $p(\boldsymbol{y}|\boldsymbol{x})$, where $\boldsymbol{x}$ is a set of input features and $\boldsymbol{y}$ is a set of output. The probability density of the target data is then represented as a linear combination of kernel functions in the form

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m(\boldsymbol{x})\phi_m(\boldsymbol{y}|\boldsymbol{x}), \tag{1}$$

where M denotes the total number of mixture components and the parameter $\alpha_m(\boldsymbol{x})$ denotes the m-th mixing coefficient of the corresponding kernel function $\phi_m(\boldsymbol{y}|\boldsymbol{x})$. Although various choices for the kernel function was possible, for this paper, we utilize the Gaussian kernel of the form

$$\phi_m(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\mu_m(\boldsymbol{x}), \sigma_m^2(\boldsymbol{x})). \tag{2}$$

Such formulation is called the Gaussian Mixture Model (GMM)-based MDN, where a MDN maps input $\boldsymbol{x}$ to the parameters of the GMM (mixing coefficient α_m, mean μ_m, and variance σ_m^2), which in turn gives a full probability density function of the output $\boldsymbol{y}$. It is important to note that the parameters of the GMM need to satisfy specific conditions in order to be valid: the mixing coefficients α_m should be positive and sum to 1; the standard deviation σ_m should be positive. The use of softmax function and exponential operator in (3) fulfills the aforementioned constraints. In addition, no extra condition is needed for the mean μ_m.

$$\alpha_m = \frac{\exp(z_m^\alpha)}{\sum_{i=1}^{M}\exp(z_i^\alpha)}, \quad \sigma_m = \exp(z_m^\sigma), \quad \mu_m = z_m^\mu \tag{3}$$

The parameters z_m^α, z_m^σ, z_m^μ are the direct outputs of the MDN corresponding to the mixture weight, variance and mean for the m-th Gaussian component in the GMM.

The objective of training the MDN is to minimize the negative log-likelihood as loss function

$$Loss = -\sum_{n} log\Big\{\sum_{m=1}^{M} \alpha_m^n(\boldsymbol{x}^n)\phi_m(\boldsymbol{y}^n|\boldsymbol{x}^n)\Big\}, \tag{4}$$

where n denotes the number of training data. The detailed derivations on closed-form gradient formulation can be found in [19], which demonstrated the capability of training the MDN using back propagation.

B. Proposed SIMP Method

Our task is to generate probability distributions of the designed semantic description given some representation of the current state. We assign a Gaussian Mixture Model (GMM) to each insertion area and multiple GMMs will be involved in one driving scenario. Each Gaussian mixture models the probability distribution of a certain type of motion for the predicted vehicle. Since obtaining the insertion location and the arriving time are the focus of our interests, a 2D Gaussian mixture is used and the predicted variables are constructed as a two dimensional vector: $\boldsymbol{y} = [y_s, y_t]^T$. The variable y_s describing the vehicle locations and the variable y_t describing the time information, can be specifically defined according to the driving environment.

Given the current state features $\boldsymbol{x}$, the probability distribution $\boldsymbol{y}_a$ over a single area a for the predicted vehicle is of the form

$$f(\boldsymbol{y}_a|\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(\boldsymbol{y}_a|\boldsymbol{\mu}_m, \Sigma_m) \tag{5}$$

with mean and covariance constructed as

$$\boldsymbol{\mu}_m = \begin{bmatrix}\mu_{s,m}\\ \mu_{t,m}\end{bmatrix}, \quad \Sigma_m = \begin{bmatrix}\sigma_{s,m}^2 & \rho_m\sigma_{s,m}\sigma_{t,m}\\ \rho_m\sigma_{s,m}\sigma_{t,m} & \sigma_{t,m}^2\end{bmatrix}, \tag{6}$$

where $\rho_m \in [-1, 1]$ is the correlation coefficient.

In addition to formulate a regression model for each insertion area, we also require the probability of entering each area for the predicted vehicle. Therefore, Deep Neural Network (DNN) was used as the basis for our Semantic-based Intention and Motion Prediction (SIMP) structure. The output of the network contains both necessary parameters for every 2D Gaussian Mixture Model (GMM) and the weight w_a for each insertion area a.

For the desired outputs, we expect not only the largest weight to be associated to the actual inserted area, but also the highest probability at the correct location and time for the output distributions of that area. Consequently, we define our loss function as

$$\begin{aligned} L = W_1\Big(-\sum_{n} log\Big\{\sum_{a=1}^{N_a} \hat{w}_a^n f(\boldsymbol{y}_a^n|\boldsymbol{x})\Big\}\Big) \\ + W_2\Big(-\sum_{n}\sum_{a=1}^{N_a} \hat{w}_a^n log(w_a^n)\Big), \end{aligned} \tag{7}$$

where N_a denotes the total number of insertion areas and $\hat{w}_a$ denotes the ground truth, which is the one-hot-encoding of the final area that the predicted vehicle entered. The last term denotes the cross-entropy loss of the area weights. Parameters $W1$ and $W2$ need to be manually tuned such that the two loss components will have the same order of magnitude during training.

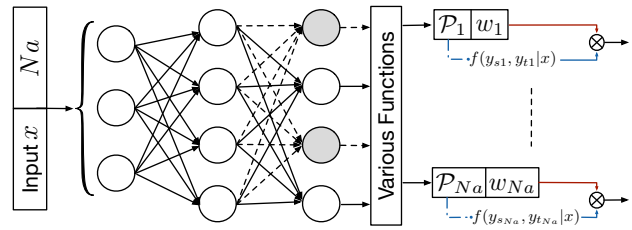


Fig. 2. Structure of the SIMP Method

The overall architecture of our SIMP method is shown in Fig. 2. Due to the first-order Markov assumption, the input features depend only on the current time step. The network consists of an input layer, several fully connected layers, and a dropout layer which ensures better generalization and prevents overfitting of the training data. After passing different types of parameters through corresponding functions, the output will satisfy the aforementioned constraints. For N_a insertion areas, the total number of output parameters can be calculated as: $N_a * (M*6+1)$. The interpretation is: there is a weight parameter w_a associated to each area $a \in N_a$, and for every $m \in M$ within an area, six parameters are needed, $\mathcal{P}_a^m = \{\alpha_m, \mu_{s,m}, \mu_{t,m}, \sigma_{s,m}, \sigma_{t,m}, \rho_m\}$, to formulate the 2D GMM.

III. An Exemplar Highway Scenario

In this section, we use an exemplar highway scenario to apply the proposed Semantic-based Intention and Motion Prediction (SIMP) method. The data source and detailed problem formulation are presented.

A. Dataset

All the data we used was taken from the NGSIM US 101 dataset which is publicly available online at [20]. It contains detailed vehicle trajectory data collected on the highway with 10 Hz sampling frequency. The measurement area is approximately 640 meters (2100 feet) in length and there are five freeway lanes plus an auxiliary lane for the on/off-ramp. For each vehicle that performs a lane change maneuver, we picked up to 40 frames (4s) before the vehicle's center intersects the lane mark; for vehicles that keep driving on the same lane for a long period, we considered these frames as input for the lane keeping maneuver. A total of 17,179 frames were selected from the dataset and split into 80% for training and 20% for testing.

B. Scenario and Problem Description

A representation of the exemplar highway driving scenario is shown in Fig. 3. The yellow car is the vehicle we decide to predict; the three blue cars (*car2, car4*, and *car6*) are the reference vehicles, which are selected as having the closest Euclidean distance to the predicted vehicle on each of the

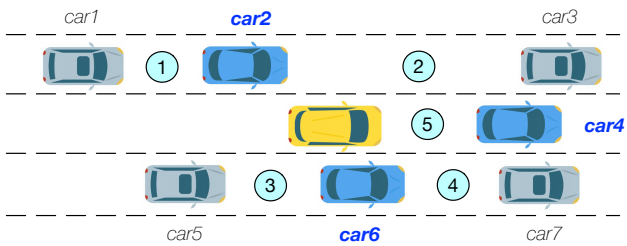


Fig. 3. An exemplar driving scenario

three lanes (we consider only the front vehicle on predicted vehicle's lane); the four gray cars (*car1, car3, car5*, and *car7*) are named as 'other vehicles', which are vehicles in front and behind each of the two reference cars: *car2* and *car6*. If any of these surrounding vehicles is too far from the predicted vehicle, we consider it as nonexistence within the range of the current scenario. Therefore, for each input frame, a maximum of three driving lanes and seven vehicles are considered.

In Fig. 3, there are five circled areas that our predicted vehicle could end up going into and we name them as Dynamic Insertion Area (DIA). If the predicted vehicle (yellow car) inserts into area 1-4, a lane change behavior is indicated; however, if it inserts into area 5, a lane keeping behavior is implied. These areas are dynamic because both their locations and sizes will vary at each time step.

In this particular highway scenario setting, the output y_s represents the absolute distance between the final insertion point and the corresponding reference vehicle for that inserted area; y_t represents the time-to-lane-change (TTLC) of the predicted vehicle. When the center of the vehicle intersects the lane mark, TTLC = 0. For the lane keeping situation, TTLC is set to a large number (4s) to represent that the vehicle has not yet decided to change the lane.

C. Features and Structure Details

For each input frame, a total of 25 input features are selected which are listed in Table I. Each input frame corresponds to 3 types of labels extracted from data: area weight, final goal location, and remaining insertion time. According to the data, the longitudinal direction is the driving direction. The current lane center (CLC) denotes the midpoint of the current lane. Because of the small angle difference between the front and the predicted vehicle, only the relative angle information for the left and right reference vehicles are considered. Time-to-collision (TTC) is calculated by dividing the speed difference by the relative distance of two vehicles. We compute the inverse of time-to-collision (iTTC) instead due to the existence of infinity TTC value as the speed difference gets close to zero.

As mentioned previously, there will be a maximum of seven cars within each input frame. If, however, a vehicle does not exist, we assign its longitudinal distance to a large number and its velocity to be the same as that of the predicted vehicle. If there is no available lane on one side of the predicted vehicle, we set the three vehicles in that nonexistent lane to be close to each other and the reference vehicle to be directly above/below the predicted vehicle. Similarly, all these three vehicles are set to have the same speed as the predicted vehicle. Such setting can guarantee the feasibility of the predicted results.

As for the network structure, we use three fully connected layers of 400 neurons each, with $tanh$ non-linear activation function. After that, a dropout layer of rate 0.5 is appended. The parameter N_a is five for this particular scenario.

TABLE I
FEATURES FOR ONE INPUT FRAME

	Feature	*Description*
Predicted Vehicle	v^y_{pred}	Absolute velocity in longitudinal direction
	$d^x_{CLC_{pred}}$	Lateral distance to the current lane center
Reference Vehicles	v^y_{ref}	Absolute velocity in longitudinal direction for every reference vehicle
	$d^y_{ref,pred}$	Relative longitudinal position between predicted vehicle and every reference vehicle
	$d^x_{(l,r),pred}$	Relative lateral position between left/right reference vehicle and predicted vehicle
	$\theta_{(l,r),pred}$	Relative angle between left/right reference vehicle and predicted vehicle
	$iTTC_{f,pred}$	Inverse time-to-collision between front reference vehicle and predicted vehicle
Other Vehicles	v^y_o	Absolute velocity in longitudinal direction
	$d^y_{o,pred}$	Position in longitudinal direction, relative to predicted vehicle
	$iTTC_{o,ref}$	Inverse time-to-collision relative to corresponding reference vehicle

IV. EVALUATION AND RESULTS

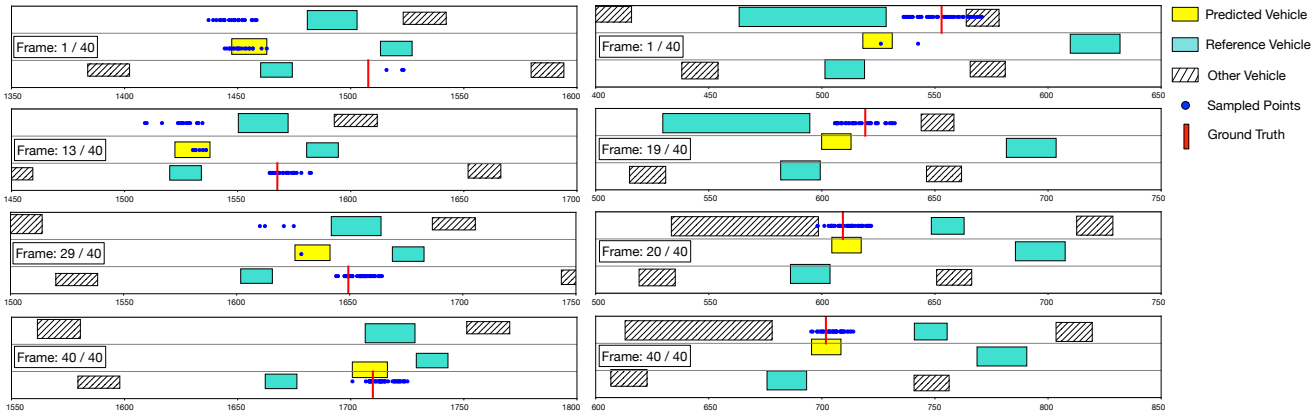
In this section, different evaluation techniques are presented to assess the model quality and the final results are discussed.

A. Evaluation Setup

1) Baseline Model: To evaluate our SIMP method, we trained a Support Vector Machine (SVM) and a Quantile Regression Forests (QRF) [21] separately. Since SVM is wildly used for classification problems, we compared it with the intention prediction part of our framework. The QRF is a combination of Quantile Regression and Random Forests [22], which extends the concept of tree ensemble learning to probabilistic prediction. Instead of point estimating the conditional mean for the selected variables like other regression methods, the objective is to estimate an arbitrary conditional quantile. The quantiles can provide detailed information of the minimum and maximum values for the dependent variable and encompass the uncertainty estimation. Hence, we compared our motion prediction part of the probabilistic framework with the QRF method for evaluation. The details of the baseline models are presented below

- SVM: kernel = (Gaussian) radial basis function (RBF)
- QRF: ntree = 1000, mtry = 5, nodesize = 10

where *ntree* is the number of trees in the forest, *mtry* is the number of random features in each tree, and *nodesize* is the



(a) Typical Lane Change (b) Sudden Change of Reference Vehicle

Fig. 4. Two example cases to visualize the performance. Fifty points were assigned to each testing frame and were sampled by following two steps: 1) multiply the total number of dots by each DIA weight. 2) for every dot assigned to each DIA, sample it according to the corresponding distribution of that area. *(The unit of the horizontal axis is in feet.)*

minimal size of terminal nodes. All these parameters were selected using five-fold cross validation.

2) Evaluation for Intention Estimation: For training and testing, each sample from our data was assigned to a semantic intention class, which is expressed as $I \in \{area1, area2, area3, area4, area5\}$. However, since these dynamic insertion areas (DIA) change constantly during the driving period, it is hard to detect the final insertion area at the early stage. Therefore, for better evaluation at the beginning of the input driving segments, we merged the original five semantic intentions into three: $\{LCL, LCR, LK\}$, where $\{area1, area2\} \in LCL$, $\{area3, area4\} \in LCR$, and $\{area5\} \in LK$. During training, the input features for SVM were the same as our method, and the labels were the corresponding final DIA numbers. The evaluation contains three steps:

i. For all testing data, create the Receiver Operating Characteristic (ROC) curve to compare our method with SVM. (Use the simplified 3 intention classes.)
ii. Find the best threshold from the ROC curve and use it to calculate the *recall, precision, F1 score* as well as the *average prediction time* for both methods.
iii. For testing data that has a TTLC smaller than the obtained *average prediction time*, analyze the performance of each DIA. (Use the original 5 semantic intention classes.)

3) Evaluation for Motion Prediction: In our problem setting, two semantic described motions are predicted: final locations in each insertion area (destination) and the remaining time to reach those locations (TTLC). For the conditional distribution of each motion, we expect not only small difference between the predicted mean and the actual value, but also centralized distribution around the predicted mean. Hence, we evaluated the *root mean squared error* (RMSE) of the output mean as well as the confidence interval for both the QRF and the SIMP method. The number of mixture components M for each DIA was set to one for analysis purpose. For the training process of QRF, we trained two separate random forest quantile regressors, where the input features remains the same and the label is either the location or the time information.

Two different intervals were chosen to assess the testing results for each method:

- SIMP-1σ: one standard deviation interval
- SIMP-2σ: two standard deviation interval
- QRF-68%: 16% to 84% quantile interval
- QRF-95%: 2.5% to 97.5% quantile interval.

B. Results and Discussion

1) Visualization of Selected Cases: We selected two distinct traffic situations to visualize our results. Each situation had 40 frames (4s) and we chose four representative frames from each case to illustrate the overall performance. [1]

A typical lane change situation is illustrated in Fig. 4(a) where the sampled points are all in the proper DIA for each frame. It is reasonable to have several possible areas at the early stage since there are multiple choices for the driver and no specific one has been chosen yet. It should be noted that it is difficult to numerically justify the correctness of these circumstances without using the human-labeled ground truth. However, as soon as the driver decides where to go, our result could be compared with the label extracted from data. We further used this case to illustrate the TTLC prediction result in Fig. 5. The differences between our resulted mean and the ground truth are all smaller than 0.3s within three seconds before lane change; besides, the predicted TTLC values for other insertion areas remain in reasonable ranges.

Since the reference vehicle will switch from one to another while the predicted vehicle is driving, we need to guarantee the capability of our method to handle such cases without large discontinuity on the prediction result. Therefore, we examined one of such cases shown in Fig. 4(b) and it can be observed that such sudden change occurs between frame 19 and 20. During this period, our sampled points are able

[1]The full video can be found on `https://www.youtube.com/watch?v=6A3Hl-mRhbI`.

to keep in the correct DIA and tightly distributed around the red target line.

2) Intention Estimation: The ROC curves of the SIMP and the SVM methods are visualized in Fig. 6. The curves were created by plotting the *true positive rate* (TPR) against the *false positive rate* (FPR) at various threshold settings. Similar to [17], we defined two positive classes (lane change left and right) and one negative class (lane keeping). The area under the ROC curve (AUC) can be used as an aggregated measure of the classifier performance. The true positive (TP) represents correct prediction of either lane change left or right, the false positive (FP) indicates mispredicting the lane change direction, and the false negative (FN) means incorrectly predicting a lane change into lane keeping.

From Fig. 6 and AUC values, we observe that our method outperforms SVM for lane change maneuvers. A threshold of 0.3 for classification was chosen for making the best trade-off between a high TPR and a low FPR. Given the selected threshold, we can further calculate the *precision* and *recall* as

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN} \tag{8}$$

and the F1 score can be obtained by the formula

$$F1 = \frac{2 * precision * recall}{precision + recall}, \tag{9}$$

which denotes how good the classification abilities are. Moreover, how early the lane change can be recognized is also in the focus of our interests. Thus, we calculated the average prediction time from the testing data that were classified as true positive. The overall performance of the two methods are compared in Table II. It is apparent from table that the proposed method has better performance than SVM in terms of both prediction accuracy and the average prediction time which describes how early the ground truth can be predicted.

Since our method can correctly forecast the predicted vehicle's intention approximately 2s in advance to the actual lane change according to Table II, we further plotted the ROC curve and calculated the AUC for each dynamic insertions area (DIA) to examine how well can SIMP predict the final insertion region. The obtained AUC values for *Area1, Area2,* and *Area3* are all equal to 1, and *Area4* has a 0.994 AUC value. The result implies that the proposed method can not only detect the lane change direction but also the specific dynamic insertion area (DIA) with high accuracy for the selected time window.

TABLE II
PERFORMANCE COMPARISON

Method	Precision	Recall	F1-Score	Avg. Predict Time (s)
SVM	0.859	0.919	0.888	1.911
SIMPF	0.936	0.925	0.931	1.957

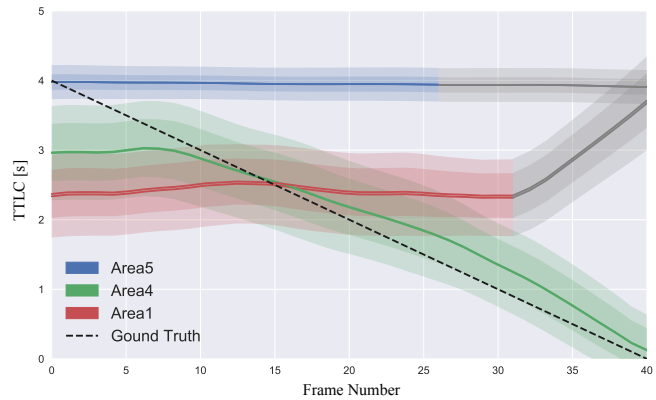


Fig. 5. TTLC illustration for the case in Fig. 4(a). We sampled 100 points from the mixture distribution of each related DIA and plotted the mean as well as the 3σ and 1σ prediction intervals for these samples. When area weight is too small to be associated with sampled points, the TTLC result of that area at the corresponding frame will be colored in gray.

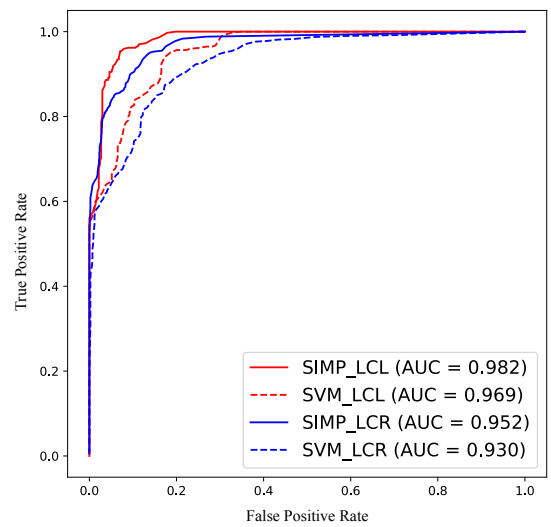


Fig. 6. ROC curve comparison.

3) Motion Prediction: The comparison results between QRF and the proposed method for two motion prediction tasks are shown in Fig. 7 and Fig. 8. The mean for QRF was obtained by calculating the 50% quantile (or median) assuming symmetric distribution. We utilized the testing data that has a TTLC smaller than the average prediction time derived in the previous section. The mean and confidence interval were calculated from the obtained output distribution of the correct insertion area. As can be seen in the plots, the RMSE of our approach for both motion predictions are smaller compared with the QRF method. The RMSE error of the TTLC tends towards zero for the lane change cases by using the SIMP method. One thing need to mention is that the error for the destination prediction is not close to zero even at $t = 0$. However, this is not unexpected given the fact that the predicted distance is relative to the reference car. Thus, the results might deviate due to the consideration of any velocity variance of the reference vehicle.

For the confidence interval comparison, it is obvious to see

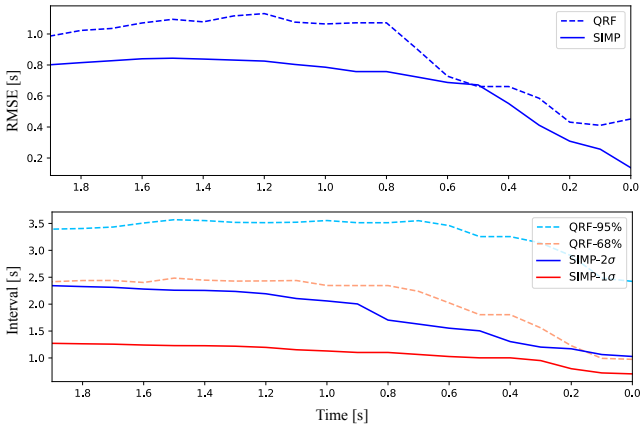


Fig. 7. Comparison of Time-to-Lane-Change (TTLC) Prediction

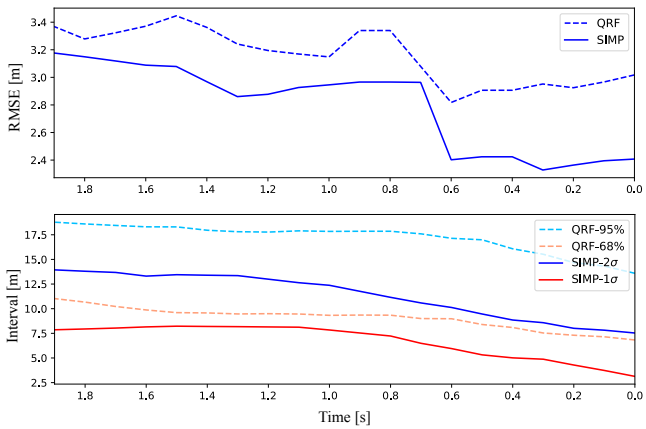


Fig. 8. Comparison of Destination Prediction

that the performance of our proposed method surpasses QRF especially for the TTLC prediction, where the 2-σ interval of the SIMP method is even smaller than the 68% interval of the QRF. The gradually decreasing difference between the one and two standard deviation interval as well as the declining interval values imply that our predicted Gaussian distribution is becoming more centralized around the ground truth as the TTLC approaching zero.

V. Conclusions

In this paper, a Semantic-based Intention and Motion Prediction (SIMP) method was proposed, which can generate various designated conditional distributions for predicted vehicles under any circumstances. An exemplar highway scenario with real-world data was used to apply the idea of SIMP. First, two representative driving cases were utilized to visualize the testing result. Then the intention prediction and the motion prediction part were separately compared with two different baseline models: SVM and QRF. Our approach outperforms these methods in terms of both the prediction error and the confidence intervals. The key conclusion is that by combining different prediction tasks using semantics in a single framework, we can easily generalize the idea into any traffic scenarios and obtain competitive performance compared to traditional methods at the same time. The output goal position and time information can be further used to generate optimal trajectories for predicted vehicles and eventually obtain a desirable path for our own autonomous vehicle. For future work, we will examine the SIMP method on more complex scenarios as well as take into account the occurrence of vehicle occlusion.

References

[1] H. M. Mandalia and D. D. Salvucci, "Using support vector machines for lane change detection," in *Proc. of the Human Factors and Ergonomics Society 49th Annual Meeting*, 2015.

[2] J. C. McCall, D. P. Wipf, M. M. Trivedi, and B. D. Rao, "Lane change intent analysis using robust operators and sparse bayesian learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 431-440, 2007.

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

[4] T. Streubel, K. H. Hoffmann, "Prediction of driver intended path at intersections," in *2014 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2014, pp. 1189-1194.

[5] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable Intention Prediction of Human Drivers at Intersections," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1665-1670, 2017.

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

[7] S. Klingelschmitt, V. Willert, and J. Eggert, "Probabilistic, discriminative maneuver estimation in generic traffic scenes using pairwise probability coupling," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1269-1276.

[8] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv:1708.06374*, 2017.

[9] S. Hoermann, D. Stumper, and K. Dietmayer, "Probabilistic long-term prediction for autonomous vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017.

[10] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *2010 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1625-1631.

[11] W. Zhan, C. Liu, C. Y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *2016 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 459-464.

[12] F. Altché, and A. De La Fortelle, "An LSTM network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC): Workshop*. IEEE, 2017.

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

[14] J. Wiest, M. Höffken, and U. Kreßel, and K. Dietmayer, "Probabilistic Trajectory Prediction with Gaussian Mixture Models," in *2012 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2015, pp 141-146.

[15] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in IROS, 2009.

[16] E. Rehder and H. Kloeden, "Goal-Directed Pedestrian Prediction", In Proceedings of 2015 IEEE International Conference on Computer Vision Workshop, pp. 139-147, 2015.

[17] H. Q. Dang, J. Fürnkranz, A. Biedermann, and M. Hoepfl, "Time-to-Lane-Change Prediction with Deep Learning," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*.

[18] C. Wissing, T. Nattermann, K. H. Glander, and T. Bertram, "Probabilistic time-to-lane-change prediction on highways," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 1452-1457.

[19] C. M. Bishop, "Mixture Density Network", 1994.

[20] U.S. Department of Transportation Intelligent Transportation Systems Joint Program Office (JPO). Avaliable: https://www.its.dot.gov/data/

[21] N. Meinshausen, "Quantile regression forests," *Journal of Machine Learning Research*, vol. 7, pp. 983-999, 2006.

[22] L. Breiman, "Random forests," *Machine Learning*, vol.45, no. 1, pp. 5-32, 2001.