Uncertainty-Aware Driver Trajectory Prediction at Urban Intersections

Xin Huang[1,2], Stephen G. McGill[1], Brian C. Williams[2], Luke Fletcher[1], Guy Rosman[1]

Abstract— Predicting the motion of a driver's vehicle is crucial for advanced driving systems, enabling detection of potential risks towards shared control between the driver and automation systems. In this paper, we propose a variational neural network approach that predicts future driver trajectory distributions for the vehicle based on multiple sensors.

Our predictor generates both a conditional variational distribution of future trajectories, as well as a confidence estimate for different time horizons. Our approach allows us to handle inherently uncertain situations, and reason about information gain from each input, as well as combine our model with additional predictors, creating a mixture of experts.

We show how to augment the variational predictor with a physics-based predictor, and based on their confidence estimations, improve overall system performance. The resulting combined model is aware of the uncertainty associated with its predictions, which can help the vehicle autonomy to make decisions with more confidence. The model is validated on real-world urban driving data collected in multiple locations. This validation demonstrates that our approach improves the prediction error of a physics-based model by 25% while successfully identifying the uncertain cases with 82% accuracy.

I. INTRODUCTION

In the case of parallel autonomy systems, a human driver and an autonomous system share control of a vehicle. When the driver's actions will put the vehicle and its surroundings at risk, the autonomous system should intervene to avoid a calamity. In order to plan alternative actions for the car, the autonomous system requires knowledge of driver intention. Trajectory prediction, then, is an important component in improving advanced driver-assistance systems (ADAS).

Several properties are required for successful and actionable prediction of a human driver's intent. Models must reason about the inherent uncertainty of the future trajectory in both the immediate and longer term. They must leverage all available sensory cues, and be able to reason about when the risk of the driver's control choices outweigh the risk of a system intervention.

Many existing trajectory prediction algorithms [1], [2] output deterministic results efficiently. However, these methods fail to capture the uncertain nature of human actions. Probabilistic predictions are very useful in many safety-critical tasks such as collision checking and risk-aware motion planning. They can express both the intrinsically uncertain

Fig. 1: Illustration of a motivating example where a vehicle is in front of an intersection. The sampled predicted trajectories using our approach are plotted in blue, where the groundtruth future trajectory is plotted in red. In parallel autonomy, the autonomous system can leverage the predicted driver trajectory to avert risk and improve the driving. This requires the system to be confident of its predicted trajectories.

prediction task at hand (human nature) and reasoning about the limitations of the prediction method (knowing when an estimate could be wrong [3]).

To incorporate uncertainties into prediction results, data-driven approaches can learn common characteristics from datasets of demonstrated trajectories [4], [5], [6]. Unfortunately, these methods often express uni-modal predictions, which may not perform well in scenarios where the driver can choose among multiple actions. For instance, the vehicle in front of the intersection in Figure 1 can move in two ways - turn left or go forward. To deal with such situations, our approach predicts multiple choices for the target vehicle.

Highway vehicle trajectory prediction [7], [8] represents a significant body of research, but work is lacking for urban driving prediction. Intersections, for instance, were responsible for 40% of crashes happened in the United States in 2008 [9]. We therefore focus on predicting trajectories for vehicles driving in urban environments. This is more challenging than highway trajectory prediction due to more complicated environments with different road shapes and dynamic objects, as well as the variety of available driving actions for the driver. Additionally, in many cases it is crucial to be aware of the confidence of the prediction. In cases where those predictions cannot be accurately made, a later planning or parallel autonomy layer can take this into account, avoiding catastrophic outcomes due to mispredictions.

Kinematic and dynamic motion models are well studied, and they are efficient for motion predictions, especially over a short horizon [1], [10]. However, due to unforeseen

changes in control inputs, these models may fail over a longer horizon. Our approach arbitrates between a wheel odometry model that works well in simple predictions and a deep neural networks model that deals with more complicated cases. We do so by introducing a learned confidence estimator that predicts the effectiveness of each prediction method for different future timepoints, conditioned on the inputs.

The input features to our problem include on-board vehicle sensor data, while foregoing mapping information. While mapping information can be incorporated into driving predictions [11], mapping data can prove unreliable in many scenarios due to changing environments such as construction areas. We choose sensor inputs with an emphasis on invariance to changing environments so that our model can generate reliable predictions. Finally, using data from affordable and widely available sensors allows our algorithms to be deployed across a spectrum of vehicle platforms.

Our work has three major contributions. First, we propose a variational trajectory prediction framework based on a deep neural network (DNN) with inputs from multiple sensor modalities, each of which contributes to different aspects of prediction, as we demonstrate. Second, we combine this variational predictor with an alternative predictor, and choose the most confident outcome using a confidence measure. The confidence measure is learned through a separately trained DNN, and helps the final predictor achieve lower prediction error. Third, we quantify the prediction uncertainty with respect to the future time horizon. Knowing how far into the future a prediction can be trusted is important for choosing between different actions in shared-autonomy settings.

The remainder of the paper is organized as follows: we review the existing work in Section I-A, and formulate our problem in Section II. We then introduce the proposed method including a DNN-based variational trajectory predictor and a DNN-based confidence estimator in Section III. Overall performance is presented in Section IV, followed by a discussion on results in Section V.

A. Related Works

Several approaches exist for shared autonomy in terms of control and interfaces. The most common approach is to use advanced driver-assistance systems (ADAS) [12], such as adaptive cruise control. Parallel autonomy vehicle shared-control frameworks, such as [13], [14], produce safe trajectories, while minimizing the deviation from driver inputs. In order to minimize this deviation (as well as gauge risk), the system needs to know where the driver is going, which is a major focus of our work.

Many approaches have been used to predict vehicle trajectories [15]. Maneuver-based models [1], [8], [16], [17] employ a hierarchical approach, estimating high-level maneuvers and predicting vehicle trajectories for each maneuver type. With delineated maneuver classes, the predicted trajectories cover multiple actions of drivers. For example, Deo et al. [8] use a maneuver-based approach to predict multi-modal trajectories, and show promising results on highway driving data. Tran et al. [17] learn maneuvers at intersections using

Gaussian regression models, and, given the most likely class of maneuver, predict trajectories. Although these approaches afford multi-modal predictions, defining and labeling maneuvers can be challenging and time-consuming – especially in urban driving with many road shapes and maneuver types.

Some prediction approaches utilize temporal networks [7], [8], [18], [19], [20] to capture the dynamics of the car and local environmental changes. For example, long short-term memory (LSTM) networks [7], [18] aid in predicting the grid-based vehicle locations probabilistically in horizons up to two seconds. Recently, [19] applies an RNN encoder-decoder framework with image and laser scan inputs to predict multi-modal trajectories in urban environments. The framework first proposes trajectory samples and then refines them with an inverse optimal control method. In comparison, our approach focuses on producing predictions directly, without using a refinement scheme. For efficiency and robustness during training and testing, we capture the vehicle's past trajectory by a set of basis functions. As we show in the results section, using the past trajectory achieves comparable performance to RNN-based techniques.

By combining a short-term motion-based predictor and a long-term maneuver-based predictor, [1] generates accurate trajectories over a horizon of four seconds. Instead of setting a time-dependent weight function, our approach chooses weights by estimated confidence over a range of prediction horizons. Our approach also allows us to gauge when the prediction result is not trustworthy, which is useful when considering whether to intervene, as in a shared-autonomy.

A crucial part of vehicle-related deep learning tasks is reasoning about the uncertainty of their outputs and knowing when the outputs are unreliable. A couple of works have addressed this concern using novelty detection on inputs. For instance, [21] uses an auto-encoder based approach to detect novel inputs based on reconstruction error for a vehicle navigation task. [22] estimates pixel-wise uncertainties from the input vehicle image using the VAE encoder network. Additionally, [23] presents a system to detect temporal and stereo inconsistencies between similar image pairs in order to recognize vehicle perception failures. In our work, we focus on producing an uncertainty-aware predictor that recognizes cases where it is not confident in its output.

II. Problem Formulation

We consider the probability distribution of the future positions of a target vehicle, conditioned on its past trajectory and current sensor inputs. These inputs include camera images, CAN bus inputs, and velocities.

We leverage pose invariance by working in the local frame of the vehicle at the predicting time. As shown in Figure 2, the x-axis points to the vehicle heading, and the y-axis points to the left of the vehicle. All geometric quantities, such as positions and velocities, are transformed to the same frame.

Similar to [5], [24], [25], we are assuming that the trajectories, projected onto some polynomial basis, form a Gaussian mixture model (GMM) with diagonal covariance matrices. Given a trajectory $\tau(t) : [0, T] \to \mathbb{R}^2$ and function basis $\mathbf{B}(t)$, the projection coefficients c can be computed
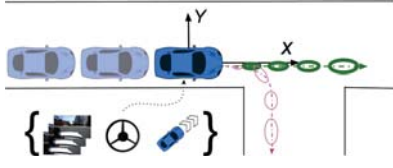
Fig. 2: Illustration of the inputs and outputs in the proposed system. The inputs include past trajectories, camera images, CAN bus data, and IMU data. The outputs are the predicted vehicle positions as a multi-component Gaussian mixture model plotted in ellipses.

with Eq. 1. Reversely, given (sampling-dependent) basis and coefficients, the trajectory can be computed in Eq. 1.

$$c = \text{Proj}_{\mathbf{B}}(\tau), \quad \tau = \mathbf{B}c. \tag{1}$$

This assume that the vehicle's localization is accurate enough, as was the case in our experiments. The basis projection operators are well defined regardless of the length of trajectory or sampling times, reducing the dimensionality of the input and output spaces for the learning problem.

The input X to our model includes

- Projected coefficients of past trajectory,
- Steering wheel angle and gas pedal position,
- Angular velocity and linear velocity,
- Images from front camera and two side cameras.

The output of the model is a probability distribution over future trajectory, which can be transformed from a set of projected coefficients. Each coefficient is represented as a Gaussian mixture model. The number of components is predefined and used to represent the number of possible movements for the target vehicle.

The goal is to estimate two conditional distributions $P(Y_G|X)$ and $P(Y_C|X)$. The first output Y_G is the GMM parameters for the variational predictor including weights w, means μ, and variances σ of projected coefficients for the future trajectory. By modeling the prediction accuracy as a function of predicting horizon, the second output Y_C contains a set of second-order polynomial coefficients that maps the time horizon to confidence scores (e.g., L2 prediction error, root mean squared error) for each candidate predictor.

III. METHOD

In this section, we describe our proposed predicting system depicted in Figure 3a. The system includes a variational trajectory predictor that outputs a set of GMM parameters, and a confidence estimator that generates confidence scores for the variational trajectory predictor as well as additional expert trajectory predictors. The mixture predictor is responsible for choosing the best predictor among candidates according to the scores estimated by the confidence estimator, or providing a warning if none of the candidates are trustworthy.

A. Variational Trajectory Predictor

The variational predictor model (see Figure 3b) is used to predict probabilistic coefficients as described in Section II. It consists of a number of children networks processing the sensor inputs and a predictor net that outputs the parameters of Gaussian mixture models.
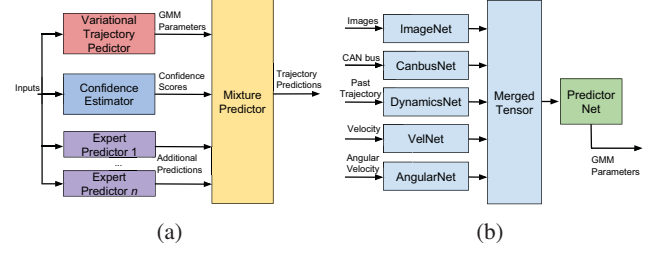

Fig. 3: Architecture diagrams of the proposed system (a) and the variational trajectory predictor (b).

1) Children Networks: Since the input data comes from various sensors and have different meaning and units, we implement a number of children networks specifically for each input, including a pre-trained VGG image network [26] to process image data, and a set of networks for the rest children, consisting of a number of fully connected layers stacked on the top of each other with ReLU activations. Between fully connected layers we use a batch normalization layer to adjust and scale activation, and a dropout layer to prevent overfitting.

The outputs of each child network are concatenated together and sent to the predictor network. We add a block-dropout scheme at the concatenation that allows each child network to be ignored with a certain probability to avoid overfitting and to allow redundancy in the overall network.

2) Predictor Network: We use a variational prediction network to further process the information from all children. It consists of stacked fully connected layers with decreasing layer width, and dropouts layers in between. The network outputs GMM parameters, including weights, means, and (diagonal) covariances for each mixture component.

B. Confidence Estimator

The confidence estimator uses a similar structure to the variational trajectory predictor. It outputs a set of coefficients for different confident scores for each candidate predictor.

C. Additional Expert Predictor

We use an odometry-based predictor as the additional expert predictor that uses a wheel odometry model [27] to compute the future positions of the vehicle by assuming the vehicle drives at constant turning rate and velocity.

Thanks to the modularity of our proposed framework, we can use more than one expert predictor. In this work, we only choose one for the sake of simplicity, and it performs well in complementing the variational predictor as discussed in Section IV.

D. Models training

Each fully connected child network in Figure 3b has two hidden layers with a dimension of 10. The predictor network has four hidden layers where each of the first three layers has a dimension of 100 and the fourth layer has a dimension of 50. A batch size of 64 is used and the model is trained for 2000 epochs using Adam with a learning rate of 0.0001. Each child network has a dropout probability of 5%. The confidence estimator network shares the same architecture and training procedures, but has a different output size and

Fig. 4: Examples of image inputs encountered in our data. Top row: stop sign with a pedestrian, unprotected left turn, driving around roundabout. Bottom row: clear day weather, rainy weather, shady weather.

is trained with a different loss function. Both networks are implemented using PyTorch and trained on an AWS server with 4 Tesla V100 GPUs in parallel.

1) Variational Trajectory Predictor Training: The model is trained as in variational inference, with the loss function defined to be negative log-likelihood of the groundtruth trajectory coefficients given GMM parameters output by our model, in addition to several regularization terms.

The log probability for a single Gaussian component with basis dimensionality D is computed using Eq. 2, and the negative log-likelihood for all mixture components is computed by Eq. 3.

$$LP_i(c) = \frac{1}{2}\sum_{d=1}^{D} -\log(2\pi\sigma_d^2) - (c_d - \mu_d)^2/\sigma_d^2 \quad (2)$$

$$\ell_{NLL}(c) = -\log\sum_i w_i \exp LP_i(c), \quad (3)$$

where c_d are the individual projection coefficients of the trajectory. To ensure the output weights and variance values are reasonable, we introduce an L2 loss on weight summation, an $L_{0.5}$ norm loss on individual weights, and an L2 loss on standard deviations. The total loss is a summation of individual losses described above.

2) Confidence Estimator Training: The loss function is defined as the L2 error between the predicted confidence scores computed using the coefficients output by our model and the actual confidence scores. The confidence score can be represented by any loss metric well-defined over the variational trajectory predictor and the expert trajectory predictor(s).

IV. EXPERIMENTAL EVALUATION

In this section, we start by describing the dataset used to train, validate, and test our model. We then show the contributions of each child network to the prediction results and describe the performance of the confidence estimator, followed by a discussion on the mixture predictor and its temporal performances.

A. Data Collection and Processing

Since we are mostly focused on intersection turning predictions, we decide to train and validate our model with driving data that are close to intersections (see Figure 4). The data was collected by a Toyota Lexus vehicle equipped with front and side cameras, Global Positions System (GPS),

Model Name	I	RMSE [m]	L2 [m]	Hard %
Variational Predictor	N/A	1.15	3.01	4.83
w/o Past Dynamics	0.81	1.29	3.37	8.43
w/o CAN bus	0.73	1.28	3.41	10.24
w/o IMU	0.85	1.31	3.45	10.47
w/o Image	0.74	1.34	3.64	15.57

TABLE I: Statistical performance of children networks. Left-to-right columns: information gain, root mean squared error, L2 endpoint error at 3 seconds, percent of hard cases.

and Inertial Measurement Unit (IMU). It consists of 14 trips with a total duration of 30 hours in Cambridge MA, Ann Arbor MI, and Los Altos CA. Different weather and lighting conditions (see Figure 4) are included in the dataset, which is partitioned into a training and validation set and a testing set. The training and validation set is split arbitrarily twice for training the variational trajectory predictor and the confidence estimator, respectively. All the metrics mentioned in this section are computed on the same testing set.

B. Children Networks

In this part, we show the complementary contributions from children networks in terms of both statistics and specific examples. In order to quantify the performance statistically, we measure the negative log-likelihood (NLL) to estimate the information gain I from the model. Information gain, or mutual information, for a specific input is given by the difference in conditional entropy with and without that input [28]. In our case this becomes the difference in log-probabilities over all data:

$$I(\tau; x) = H(\tau|X^-) - H(\tau|X^-, x) \quad (4)$$
$$= -\sum_{\tau, X^-, x} p(\tau, X^-, x)\left[\log p(\tau|X^-) - \log p(\tau|X^-, x)\right]$$

where $H(\cdot)$ is the entropy, x denotes an input of interest, X^- denotes the rest of the inputs, and $\log p(\tau|X^-, x)$, $\log p(\tau|X^-)$ are obtained by networks trained with or without x as an input.

In addition, we use root mean squared error (RMSE) to show the prediction quality along the entire future trajectory and L2 error at the predicting horizon of 3 seconds. To understand when the model fails catastrophically, we compute the percentage that the output predictions have an L2 error greater than 5 meters. The results are recorded in Table I.

1) Dynamics Network: We start by removing the dynamics network from the variational predictor model. Since we are using a second-order polynomial basis, the past coefficients capture the dynamics including the average past velocity and acceleration of the vehicle in each dimension. To check the contribution of the dynamics network, we visualize a few examples in Figure 5a where the prediction error increases drastically after removing the dynamics network from the variational predictor. We see that without knowing the past dynamics, the predictions can be slightly off when a vehicle drives smoothly on the road.

2) RNN without Dynamics Network: As stated in Section I-A, many use RNN-based approaches to learn temporal information such as dynamics from past observations. We implement an RNN model that takes the same input except

(a) Without dynamics inputs, the predicted trajectories overshoot even when the vehicle drives on a straight road.

(b) Without CAN bus inputs, there is always motion predicted even when the vehicle is not moving.

(c) Without image inputs, the predicted trajectories go out of road boundaries especially at turns.

(d) With all sensor inputs, the prediction can still be hard in less familiar or out-of-distribution scenarios.
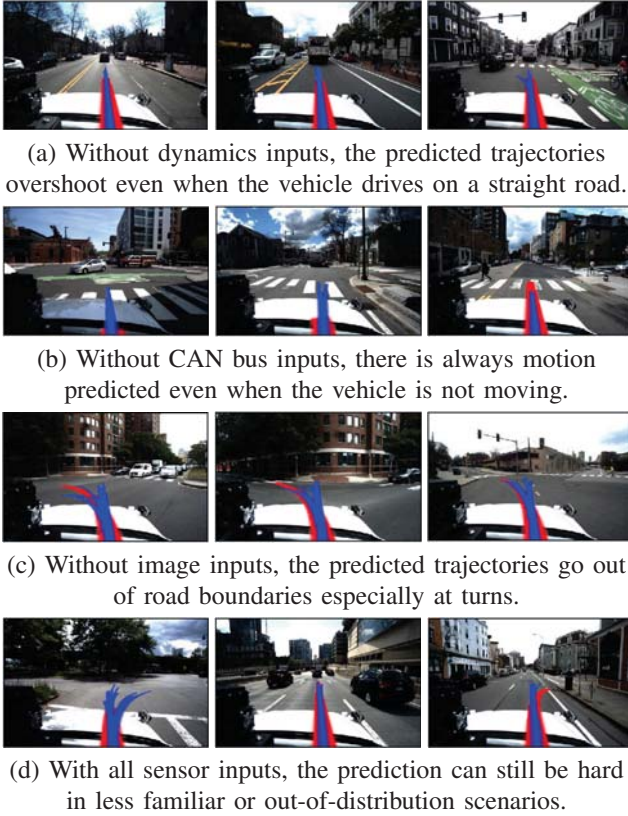
Fig. 5: Illustration of challenging examples for the variational predictor while withholding different inputs. The groundtruth future trajectory is in red, and the sampled predicted trajectories using our approach are in blue.

for the information on the past trajectory. The model achieves an information gain of 0.17 as compared to the variational network without past dynamics, which is worse than the information gain of 0.81 we would gain from the past dynamics. This indicates that the dynamics network captures the majority of the signal that RNN-based models can learn. Given that the past trajectory is usually available for vehicles with GPS sensors, we decide to use a single shot model for our variational predictor, which takes less time to converge during training time than an RNN model. We therefore do not show an RNN-based approach in the rest of the experiments and in Table I.

3) CANBUS Network: CAN bus data includes steering wheel angle and gas pedal value, which can inform the variational predictor about the instant acceleration and turning rate. From Table I, we see that after dropping the CANBUS network, the L2 prediction error increases by 13.29%. Without a CANBUS network, the variational predictor would not able to know the change in acceleration. Examples in Figure 5b show that at intersections, it is important to know the acceleration to avoid overshooting or undershooting.

4) IMU Network: In complementary to the CANBUS network, the IMU network provides information about the instant velocity and angular rate of the vehicle.

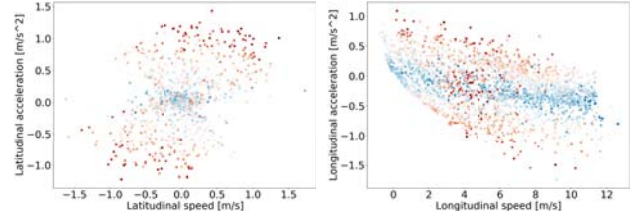*5) Image Network:* The large performance drop after removing the image network suggests that it contributes

Fig. 6: Scatter plots of the L2 error differences between our variational trajectory predictor and the odometry predictor over 3 seconds. Red means the variational predictor performs better. Blue means the odometry predictor performs better. Left: Error differences as a function of latitudinal vehicle velocity and acceleration. Right: Error differences as a function of longitudinal vehicle velocity and acceleration. For qualitative results, please refer to our video demonstration.

the most to the network. For example, the percentage of hard cases has increased by two times if we remove the image network, which means that the image network is able to reduce many hard cases by using its information on what exists in the future, especially in cases where the past dynamics or the CAN bus/IMU data is unavailable. More specifically, the pre-trained VGG network allows us to identify the upcoming environment conditions such as road shapes and obstacles. This can help to fine-tune the predicted trajectories to be close to the road curve and yield higher accuracy. As shown in Figure 5c, without an image network, the variational predictor has a hard time getting accurate predictions when the vehicle turns at intersections.

6) Variational Trajectory Predictor: By using all the children networks, the variational predictor gets information about the environment ahead of the vehicle as well as the vehicle state including acceleration, velocity, angular rate, and past dynamics. It is able to predict trajectories with an average prediction error of 3.01 meters in complicated urban driving environments.

Despite the success of the variational predictor, it can fail. Defining failures as estimation errors greater than 5 meters over the horizon, we have a 4.83% failure rate in urban driving conditions. Figure 5d shows our failure examples in challenging scenarios including places without well-defined road boundary, in an unfamiliar highway environment, and in front of a street hidden behind the buildings. For a trajectory predictor to be trustworthy in a shared-autonomy framework, it has to recognize these uncertain cases, which motivates our confidence estimator, as described in the next section.

C. Confidence Estimator

We use the results from the two candidate trajectory predictors to obtain confidence scores for each data sample. In this work, we choose to use the L2 error between the predicted and groundtruth position as our confidence score. The error distributions at predicting horizon of 3 seconds for two candidate predictors are shown in Figure 7a, with an average of 3.01 meters and 3.11 meters, respectively. In order to compare the performance between the two candidate predictors at different scenes, we show in Figure 6 that our variational predictor tends to have lower L2 errors than the
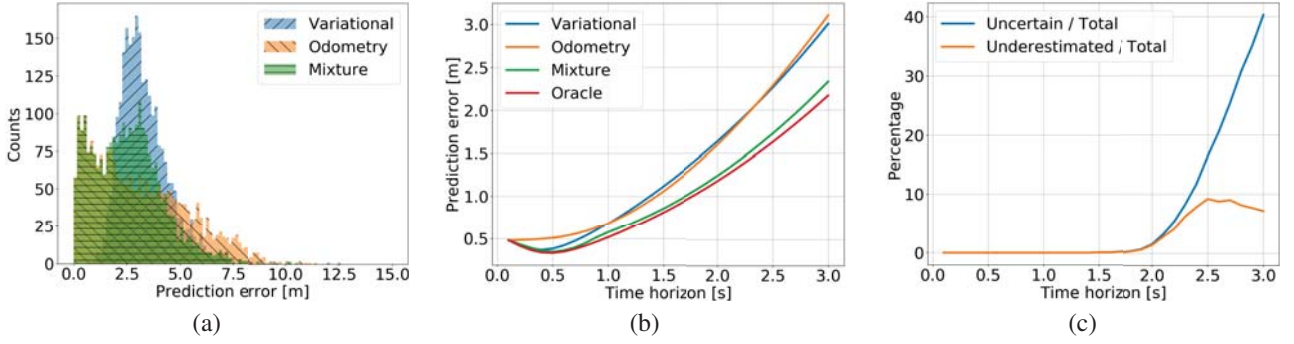
Fig. 7: Illustrations of predictors performances on the testing set. (a) Histograms of prediction errors for the variational predictor, the odometry predictor, and the mixture predictor at 3 seconds. (b) Prediction errors of three predictors and an oracle predictor, over different time horizons from 0.1 to 3 seconds. (c) Percentage of uncertain cases (cf. text) and underestimated uncertain cases over different time horizons.

odometry predictor in complicated cases where longitudinal and latitudinal accelerations are large.

The obtained confidence scores are used to train a confidence estimator that produces estimated confidence scores on input data for each trajectory predictor. The confidence estimator estimates scores for the variational predictor and the odometry predictor with an average error of 1.07 meters and 2.14 meters, respectively. As we wish to leverage the confidence estimator to choose the better predictor, we compute the percentage that it finds the correct better trajectory predictor to be 75%.

Since the driving domain is very safety-critical, we measure the frequency at which our confidence estimator underestimates the uncertain cases. We define the uncertain case as where both predictors produce errors above 2.54 meters, which is the mean error of a set of complicated cases gathered manually from the data. The percentage of underestimated samples out of all uncertain samples is 18%, which means our confidence estimator knows 82% of the time when it is outputting uncertain results. This is crucial in safety-critical systems, especially in shared-control and parallel autonomy systems, as we can change the system's actions according to the certainty of its own perception subsystem.

In comparison, in the cases where the prediction error is less than 2.54 meters, our confidence measure undershoots the error in 94% of the cases, which means we can trust the confidence estimator 89.2% of the time with 40% of uncertain cases and 60% of certain cases.

D. Mixture Predictor

The mixture-of-experts predictor arbitrates between the variational predictor and the odometry predictor, by generating their estimated confidence scores, and picking the prediction that has a higher score. The mixture-based errors are distributed in green as shown in Figure 7a with an average of 2.33 meters, which is improved by 23% compared to the average error of the variational predictor and 25% compared to the average error of the odometry predictor. The *regret* is 0.16 meters, which is defined in comparison to an *oracle* arbitrator that knows which prediction is better at every sample.

E. Temporal Behaviors

An important aspect of predictors is their behaviors as a function of the prediction horizon. We are interested in seeing the performance of the mixture predictor, and gather the mean of L2 prediction errors over a predicting horizon from 0.1 seconds to 3 seconds (see Figure 7b). The gap between the mixture predictor curve and candidate predictor curves indicates that it successfully chooses the good part from them. This can be validated by the temporally-dependent regret as illustrated by the gap between the mixture predictor curve and the oracle predictor curve. Similar to IV-C, we compute the percentage of uncertain cases over different time horizons (see Figure 7c). Starting at 1.4 seconds, the number of uncertain cases starts to increase quickly, due to the uncertainty increase in driver actions. The percentage of underestimated uncertain cases over total cases never exceeds 10%, which means that our mixture predictor is aware most of the time when it is generating predictions with large uncertainty. The underestimated curve starts to fall after 2.5 seconds, because our confidence estimator tends to output larger error estimates over longer horizons and thus is less likely to underestimate the uncertain cases.

V. CONCLUSIONS

This paper introduces an uncertainty-aware prediction method for driver trajectory prediction by augmenting a variational deep neural network trajectory predictor with a physics-based predictor using confidence estimates. The variational predictor generates probabilistic predictions from a number of aggregated children networks that process different vehicle sensor data, and the physics-based predictor acts as an additional expert predictor that provides more accurate predictions in simple cases where vehicle accelerations are small. We use a second network to estimate the uncertainty of predictions by different candidate predictors as a function of the predicting horizon. The uncertainty estimates not only help the mixture predictor achieve better accuracy over a horizon up to 3 seconds, but also provide insights on when to use our predictor in shared-autonomy. Given the safety critical nature of autonomous driving tasks, we believe our uncertainty-aware method provides a new direction for making a robust and trustworthy prediction system.

REFERENCES

[1] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 4363–4369.

[2] H. Woo, Y. Ji, H. Kono, Y. Tamura, Y. Kuroda, T. Sugano, Y. Yamamoto, A. Yamashita, and H. Asama, "Lane-change detection based on vehicle-trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1109–1116, 2017.

[3] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Advances in neural information processing systems (NIPS)*, 2017, pp. 5574–5584.

[4] D. Vasquez, T. Fraichard, and C. Laugier, "Growing hidden markov models: An incremental tool for learning and predicting human and vehicle motion," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1486–1506, 2009.

[5] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 141–146.

[6] C. Hermes, C. Wohler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," in *Intelligent Vehicles Symposium (IV), 2009 IEEE*. IEEE, 2009, pp. 652–657.

[7] S. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," *arXiv preprint arXiv:1802.06338*, 2018.

[8] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," *arXiv preprint arXiv:1805.05499*, 2018.

[9] E. Choi, "Crash factors in intersection-related crashes: An on-scene perspective (no. dot hs 811 366)," *US DOT National Highway Traffic Safety Administration*, 2010.

[10] A. Berthelot, A. Tamke, T. Dang, and G. Breuel, "Handling uncertainties in criticality assessment," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 571–576.

[11] A. Amini, G. Rosman, S. Karaman, and D. Rus, "Variational end-to-end navigation and localization," *arXiv preprint arXiv:1811.10119*, 2018.

[12] J. Pérez, D. Gonzalez, and V. Milanés, "Vehicle control in ADAS applications: State of the art," *Intelligent Transport Systems: Technologies and Applications*, p. 206, 2015.

[13] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1928–1935.

[14] F. Naser, D. Dorhout, S. Proulx, S. D. Pendleton, H. Andersen, W. Schwarting, L. Paull, J. Alonso-Mora, M. H. Ang, S. Karaman *et al.*, "A parallel autonomy research platform," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 933–940.

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

[16] M. Schreier, V. Willert, and J. Adamy, "An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2751–2766, 2016.

[17] Q. Tran and J. Firl, "Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression," in *Intelligent Vehicles Symposium (IV), 2014 IEEE*. IEEE, 2014, pp. 918–923.

[18] B. Kim, C. M. Kang, S. H. Lee, H. Chae, J. Kim, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," *arXiv preprint arXiv:1704.07049*, 2017.

[19] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 336–345.

[20] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971.

[21] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Robotics: Science and Systems Foundation (RSS)*, 2017.

[22] A. Amini, W. Schwarting, G. Rosman, B. Araki, S. Karaman, and D. Rus, "Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[23] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, "Failing to learn: Autonomously identifying perception failures for self-driving cars," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3860–3867, 2018.

[24] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? a unified framework for maneuver classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129–140, 2018.

[25] F. Havlak and M. Campbell, "Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 461–474, 2014.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] B. Goldfain, P. Drews, C. You, M. Barulic, O. Velev, P. Tsiotras, and J. M. Rehg, "Autorally an open platform for aggressive autonomous driving," *arXiv preprint arXiv:1806.00678*, 2018.

[28] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006.