Uncertainty-aware Short-term Motion Prediction of Traffic Actors for Autonomous Driving

Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen,
Fang-Chieh Chou, Tsung-Han Lin, Nitin Singh, Jeff Schneider

Uber ATG

Background and motivation

Trajectory prediction for autonomous driving with rasterization and CNN

- To ensure safe and efficient operations, an autonomous vehicle needs to accurately predict the future motions of the traffic actors in its surroundings
- We propose to rasterize high-definition maps and surroundings of each vehicle in vicinity of self-driving vehicle (SDV), thus providing complete context and information necessary for accurate prediction of future trajectory
- We trained deep CNN called **RasterNet** to predict short-term vehicle trajectories, while accounting for an inherent uncertainty of the traffic motion
- Large-scale evaluation on real-world data showed the system provides accurate predictions and well-calibrated uncertainties, indicating its practical benefits
- Following extensive offline testing, the system was **successfully tested onboard self-driving vehicles**

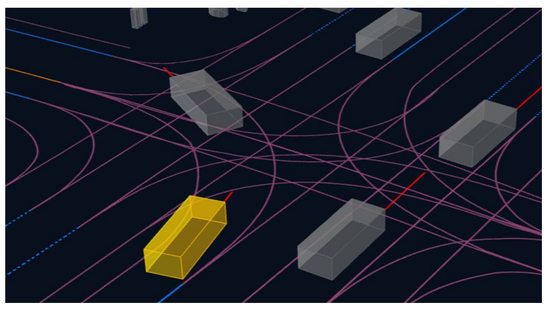


Fig 1. Complex scene seen in an internal viewer

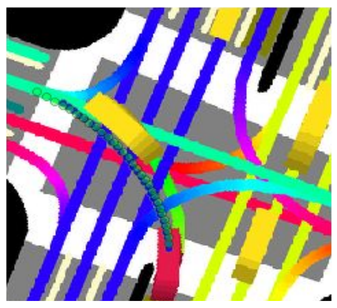


Fig 2. RasterNet input raster and output trajectory

Proposed approach

- Given info describing the scene around an actor and their recent history, the task is to predict actor movement as well as its movement uncertainty
- In the following we define the loss function used to train the CNN
- For an actor *i* observed at time *j*, and assuming historical states of all actors S_j, map data *M*, and model parameters *Θ*, define prediction error at horizon *h* as an L2-displacement between predicted and ground-truth *x/y* location

$$d_{i(j+h)} = \Big(\big(x_{i(j+h)} - \hat{x}_{i(j+h)}(\mathcal{S}_j, \mathcal{M}, \theta)\big)^2 + \big(y_{i(j+h)} - \hat{y}_{i(j+h)}(\mathcal{S}_j, \mathcal{M}, \theta)\big)^2 \Big)^{1/2}$$

- Let us assume the displacement loss comes from a half-Gaussian distribution

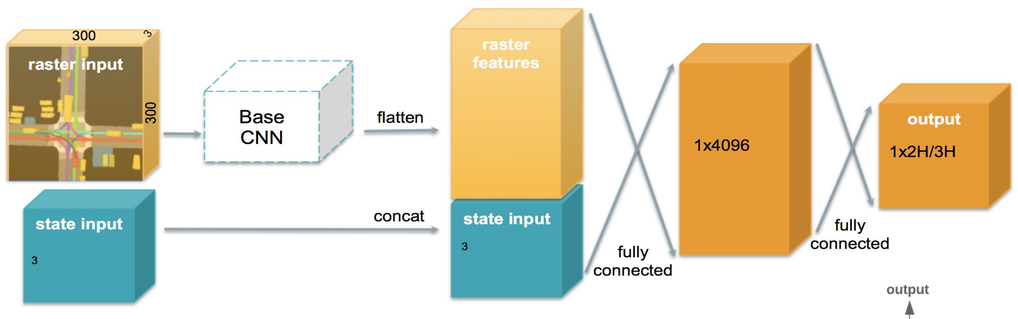
$$d_{i(j+h)} \sim \mathcal{FN}\big(0, \hat{\sigma}_{i(j+h)}(\mathcal{S}_j, \mathcal{M}, \theta)^2\big)$$

- Then, per-actor loss is a negative log-likelihood computed over all horizons *H*

$$L_{ij} = \sum_{h=1}^{H} \Big(\frac{d_{i(j+h)}^2}{2\ \hat{\sigma}_{i(j+h)}(\mathcal{S}_j, \mathcal{M}, \theta)^2} + \log \hat{\sigma}_{i(j+h)}(\mathcal{S}_j, \mathcal{M}, \theta) \Big)$$

Network architecture

- We train a CNN model to minimize and solve the above problem
- The proposed architecture is given below, outputting *x/y*-locations (total of *2H* outputs), and standard deviation for each location (additional *H* outputs)



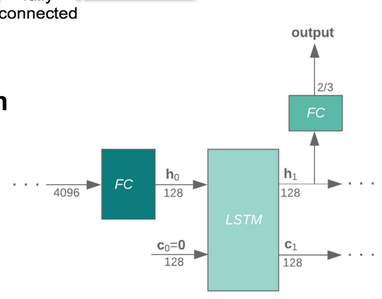


Fig 3. The proposed RasterNet architecture with a feed-forward, fully-connected trajectory decoder (top); an alternative LSTM-based trajectory decoder (right-hand image)

Future/follow-up work

- Guarantee physical realism of predicted trajectories using vehicle models
- Model interaction between actors
- Learn better trajectory loss function with GANs
- Include explicit constraints on trajectory prediction (e.g. lane following, avoiding static obstacles)
- Combine models of different actor types
- Joint Perception-Prediction models for faster inference

Empirical evaluation

Experimental setup

- We collected 240 hours of data by manually driving SDV in Pittsburgh, PA and Phoenix, AZ in various traffic conditions (e.g., varying times of day, days of the week), with collection rate of 10Hz, the same frequency the Unscented Kalman Filter (UKF) tracker was run on
- Each actor at each discrete tracking time step amounts to one data point, with overall data comprising 7.8 million examples after removing static actors
- We considered horizon of 3s (i.e., we set H = 30), and used 3:1:1 split to obtain train/validation/test data

Table 1. Comparison of average prediction errors for competing methods (in meters)

Method	Raster	State	Loss	Displacement	Along-track	Cross-track
UKF	–	yes	–	1.46	1.21	0.57
Linear model	–	yes	(2)	1.19	1.03	0.43
Lane-assoc	–	yes	–	1.09	1.09	0.19
AlexNet	w/o fading	no	(2)	3.14	3.11	0.35
AlexNet	w/ fading	no	(2)	1.24	1.23	0.22
AlexNet	w/o fading	yes	(2)	0.97	0.94	0.21
AlexNet	w/ fading	yes	(2)	0.86	0.83	0.20
VGG-19	w/ fading	yes	(2)	0.77	0.75	0.19
ResNet-50	w/ fading	yes	(2)	0.76	0.74	0.18
MobileNet-v2	w/ fading	yes	(2)	0.73	0.70	0.18
MobileNet-v2	w/ fading	yes	(5)	0.71	0.68	0.18
MobileNet-v2 LSTM	w/ fading	yes	(5)	**0.62**	**0.60**	**0.14**

How well-calibrated are the uncertainty estimates?

- We used reliability diagrams to quantify the calibration (note we are somewhat under-confident at 1s)

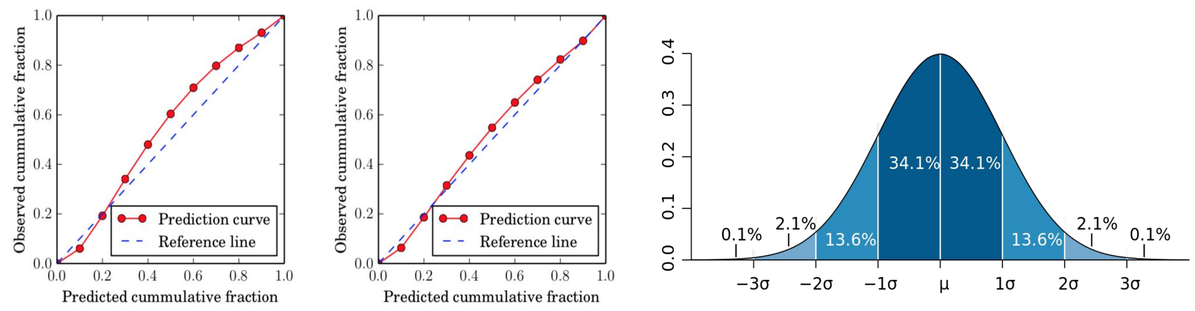


Fig 4. Reliability diagrams at horizons of 1s and 3s and the illustration of Gaussian distribution

Case studies

- In Figure 5 we give analysis of three scenes commonly encountered in traffic. As we see, the model provided accurate short-term trajectories, as well as reasonable and intuitive uncertainty estimates

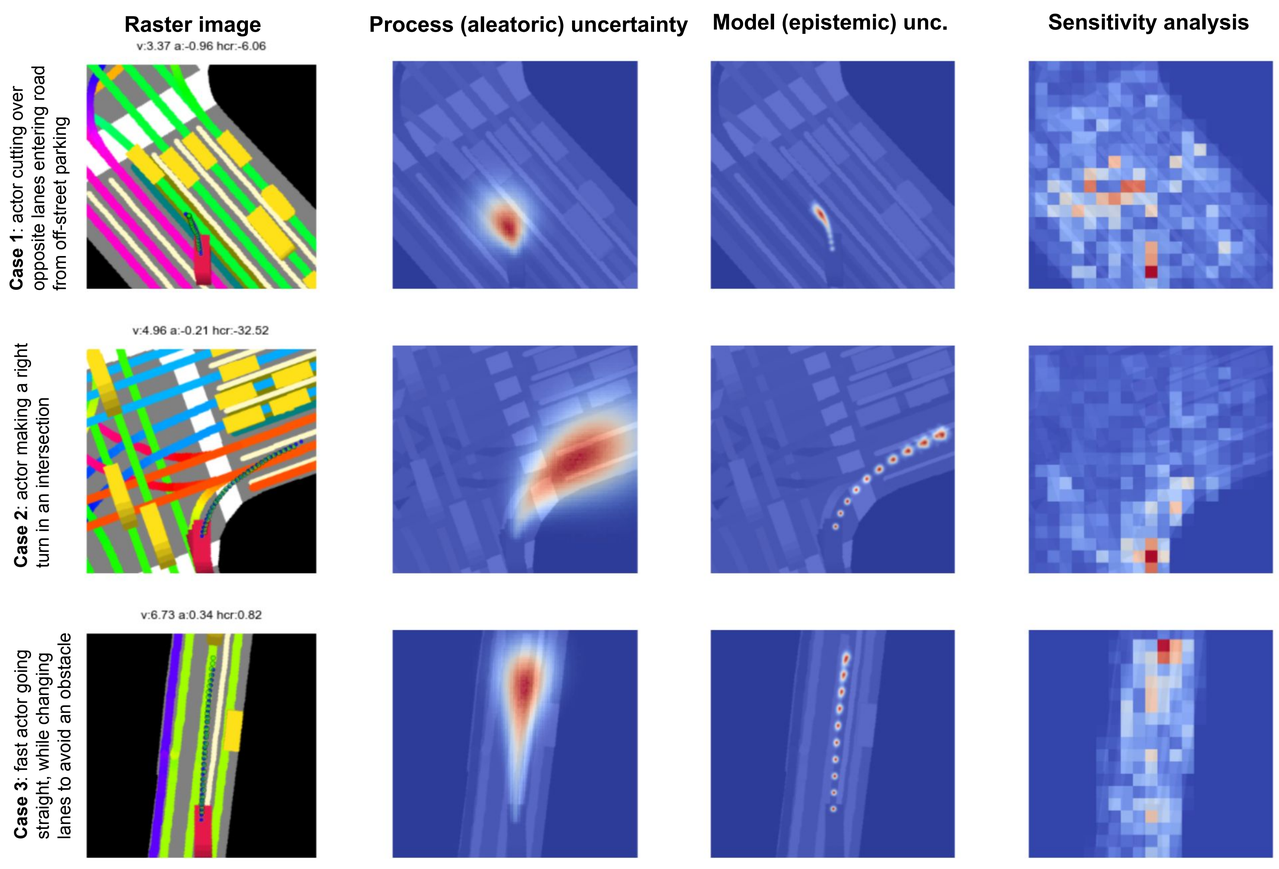


Fig 5. Analysis of the RasterNet model for three case studies

In-depth error analysis using error heatmaps

- In Figure 6 we provide an additional analysis of cross- and along-track errors
- At each timestamp of the event (*x*-axis), we color-code errors at each prediction horizon up to 3 seconds in the future (*y*-axis)
- The actor starts to approach the intersection at around 1s mark, and initiates the turn at around 3s mark
- While both methods initially do not predict the turn, MN-v2 is faster to capture the new behavior

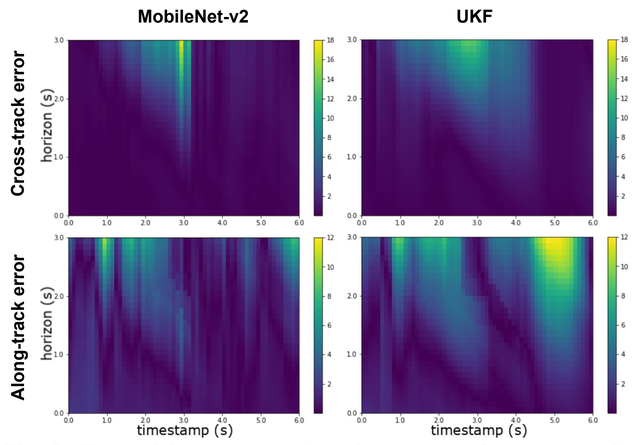


Fig 6. Error histograms for the second case study