Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks

Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin,
Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, Nemanja Djuric

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 surrounding.
- Our prior work RasterNet [1] tackles this problem with scene rasterization and deep convnets. RasterNet combines the output of an existing detection and tracking system (objects with state estimates, e.g. positions, bounding boxes, velocities), with HD-map info (locations of lanes and crosswalks, lane directions, etc.) into an object-centric raster input image for each object. A CNN is then applied to predict the future positions of the target objects.

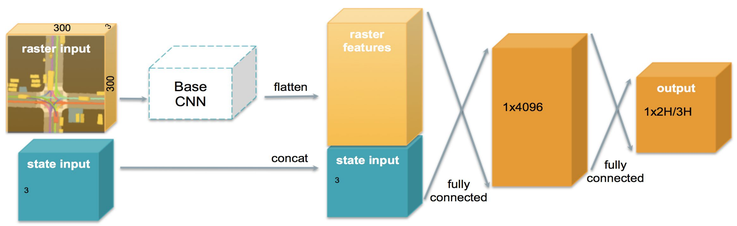

Fig 1. RasterNet architecture

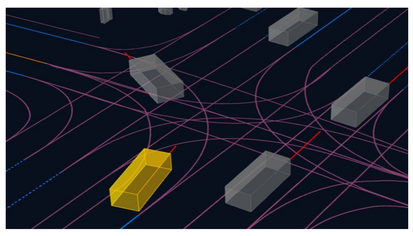

Fig 2. Complex scene in an internal 3D viewer

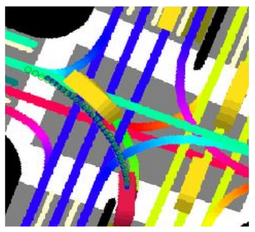

Fig 3. RasterNet input raster and output trajectory

The challenge of multimodality

- The future is inherently multi-modal
- A single-modal model often outputs the average of multiple modes
- A better model should output multiple trajectories along with their probabilities

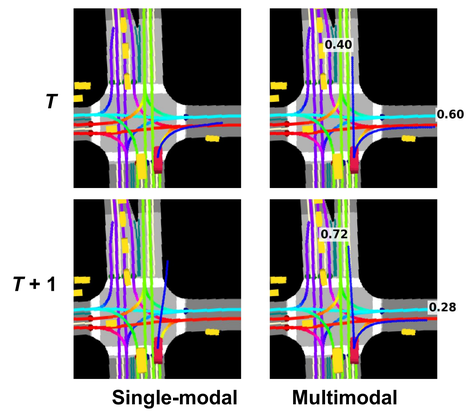

Fig 4. Comparison of uni- and multi-modal results

Reference

[1] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, and J. Schneider. Short-term motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1808.05819, 2018*.

Methods

Network architecture

- We used the existing RasterNet architecture and extended the network head to output *M* trajectories τ_m each with probability p_m

Multimodal loss function

- Find the trajectory τ_{m^*} that is *closest* to the ground-truth τ_{gt}

$$m^* = \text{argmin dist}(\tau_m, \tau_{gt})$$

- A trajectory L2 regression loss with only the matching trajectory τ_{m^*}
- A classification loss as the cross-entropy between p and $p_{gt} = \text{one\_hot}(m^*)$
- The total loss is the summation of the regression loss and the classification loss

$$\mathcal{L} = \mathcal{L}_{traj}(\tau_{m^*}, \tau_{gt}) + \mathcal{L}_{cls}(p, p_{gt})$$

- The total loss is the summation of the regression loss and the classification loss

Mode selection policies for deciding m*

- The simplest option: *min_displacement*
  - Could cause undesired mode switching problems illustrated below
- Improved selection logic: *min_angle*
  - Pick the mode that has the the smallest angle from the ground-truth

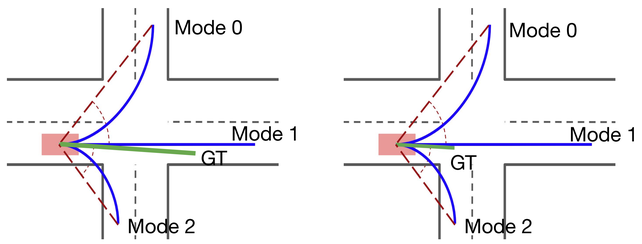

Fig 5. Illustration of *min_disp* and *min_angle* policies

Multimodal prediction with goal path rasterization

- Another approach that we propose for handling the multimodal prediction problem is to encode the goal path as an additional element in the image rasterization (painted in pink). And the model is also able to output diverse trajectories.

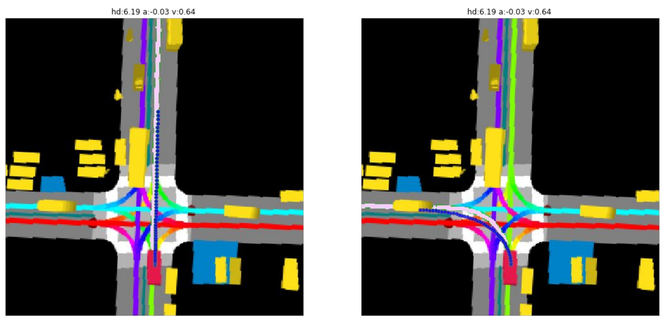

Fig 6. Example outputs for straight and left-turning goal path

Results

Quantitative result

- Our approach successfully produces diverse trajectories with probabilities.

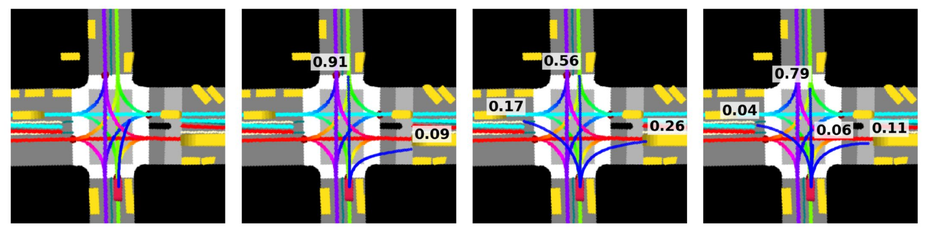

Fig 7. Example inference for models with increasing modality *M*

Trajectory prediction errors

- Showing minimal error among all modes with ≥ 0.2 probability
- Our approach produces diverse trajectories with lower errors
- Compare with Unscented Kalman Filter (UKF) tracker, single-mode trajectory predictor (STP), Mixture of experts (ME), Mixture Density Net (MDN), the proposed multi-modal predictor (MTP)

Table 1. Comparison of various methods

Method	No. modes	Displacement [m] @1s	Displacement [m] @6s	Displacement [m] Average	Along-track [m] @1s	Along-track [m] @6s	Along-track [m] Average	Cross-track [m] @1s	Cross-track [m] @6s	Cross-track [m] Average
UKF	1	0.54	10.58	3.99	0.41	7.88	2.94	0.26	5.05	1.94
STP	1	0.34	4.14	1.54	0.30	3.91	1.45	0.11	0.72	0.30
ME	2	0.34	4.17	1.55	0.30	3.93	1.45	0.11	0.74	0.30
ME	3	0.34	4.13	1.54	0.30	3.90	1.44	0.11	0.72	0.30
ME	4	0.34	4.13	1.54	0.30	3.89	1.44	0.11	0.73	0.30
MDN	2	0.37	4.18	1.58	0.33	3.86	1.46	0.10	0.61	0.27
MDN	3	0.27	3.31	1.26	0.22	2.95	1.12	**0.09**	0.60	0.26
MDN	4	0.27	2.91	1.18	0.21	2.55	1.05	**0.09**	0.59	0.26
MTP w/ disp.	2	0.25	2.72	1.07	0.20	2.49	0.97	**0.09**	0.49	**0.23**
MTP w/ disp.	3	**0.23**	**2.31**	**0.94**	**0.17**	**2.05**	**0.82**	**0.09**	**0.46**	**0.23**
MTP w/ disp.	4	0.24	2.51	1.00	0.18	2.23	0.88	**0.09**	0.51	0.24
MTP w/ angle	2	0.26	2.80	1.10	0.21	2.56	1.00	**0.09**	0.51	0.24
MTP w/ angle	3	**0.23**	2.33	0.95	**0.17**	2.08	0.84	**0.09**	**0.46**	**0.23**
MTP w/ angle	4	0.25	2.57	1.03	0.19	2.29	0.91	**0.09**	0.51	0.24

min_displacement vs. min_angle

- *min_angle* reduces the errors for left/right turning actors
- Improved handling of actors in intersections

Table 2. Comparison of mode selection policies

Predictor	No. modes	Displacement @6s [m] Left-turn	Displacement @6s [m] Straight	Displacement @6s [m] Right-turn
MTP w/ disp.	2	4.48	2.59	5.64
MTP w/ disp.	3	4.18	**2.18**	5.42
MTP w/ disp.	4	4.51	2.37	5.47
MTP w/ angle	2	4.65	2.67	5.66
MTP w/ angle	3	**4.10**	2.21	**5.17**
MTP w/ angle	4	4.91	2.43	5.52

Mode probabilities

- The learned mode probabilities are well-calibrated

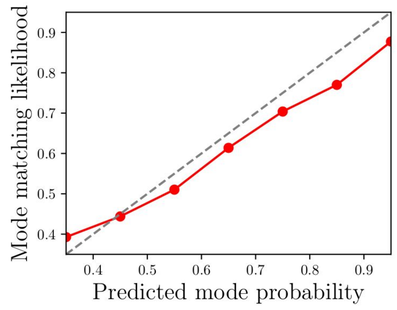

Fig 8. Analysis of mode probability calibration