Benchmarking Behavior Prediction Models in Gap Acceptance Scenarios

Supplementary Materials - Implementation Details

Julian F. Schumann, Jens Kober, Arkady Zgonnikov

I. Datasets

Irrespective of the dataset, the following implementations are used:

- $a_{\text{brake}} = 4\,\mathrm{ms}^{-2}$, $\delta t = 0.2\,\mathrm{s}$, $t_\epsilon = 0.01\,\mathrm{s}$, and $n_p = 100$.
- To get Δt, one maximizes the term

$$\min\left\{N_A,\ N_{\overline{A}}\right\} \tag{1}$$

with

$$N_A = \sum_i a_i\,,\ N_{\overline{A}} = \sum_i (1 - a_i)\,. \tag{2}$$

- When additional agents V_i are unavailable, vehicles in this role are created. These are placed far away from the ego and target vehicles, so their influence on the prediction model should be minimal.

A. Lane changes

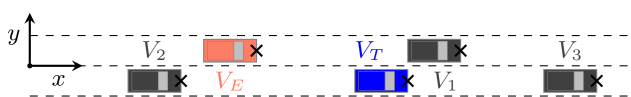

Fig. 1. The lane change scenario. The black crosses symbolize the position vector $\boldsymbol{x}_i$ belonging to agent V_i.

The first dataset—called highD—is a natural driving set recorded on German highways, captured by using drones [1], which entails lane changes of V_T towards a faster lane to the left. Here, a gap is accepted when V_T merges in front of V_E. The trajectories of additional surrounding vehicles are also recorded (or their existence is assumed outside the camera's scope). These are V_1 in front of V_E as well as V_2 behind and V_3 in front of V_T (Fig. 1).

The following conditions

$$\begin{aligned}
C_S(t) &: x_1(t) - x_T(t) = 5\,\mathrm{m} \wedge \dot{x}_1(t) > \dot{x}_T(t)\\
C_C(t) &: x_T(t) - x_E(t) = 5\,\mathrm{m}\\
C_A(t) &: y_T(t) = 0\,\mathrm{m} \wedge \dot{y}_T(t) > 0
\end{aligned}$$

are now used for determining the characteristic time-points, with

$$\begin{aligned}
t_{\widehat{C}}(t) &= \frac{x_T(t) - x_E(t) - 5\,\mathrm{m}}{\max\left\{\dot{x}_E(t) - \dot{x}_T(t), 0\right\}} + t\\
t_{\text{brake}}(t) &= \frac{1}{2}\frac{\max\left\{\dot{x}_E(t) - \dot{x}_T(t), 0\right\}}{a_{\text{brake}}}\\
\Delta t &= 11.8\,\mathrm{s}\,.
\end{aligned}$$

When assembling the restricted version of this dataset, only samples satisfying on of the following two requirements are included:

- V_T changes the lane after V_E has overtaken it, indicating that V_T has already been looking out for feasible gaps before but has rejected the one offered by V_E. 1025 rejected gaps fulfill this condition.
- V_T is substantially faster than V_3, i.e., $t_{\widehat{C}}(t_S) - t_S \geq 2\left(t_{\widehat{3}}(t_S) - t_S\right)$:

$$t_{\widehat{3}}(t) = \frac{x_3(t) - x_T(t) - 5\,\mathrm{m}}{\max\left\{\dot{x}_T(t) - \dot{x}_3(t), 0\right\}} + t \tag{3}$$

This condition is satisfied for 133 rejected gaps where it is suggested that V_T has consciously chosen to brake instead of accepting the gap.

As those conditions are not exclusive, only 1075 rejected gaps are included in the final dataset (Table **??**), for which the Equation (1) results in $\Delta t = 8.7\,\mathrm{s}$.

B. Roundabouts

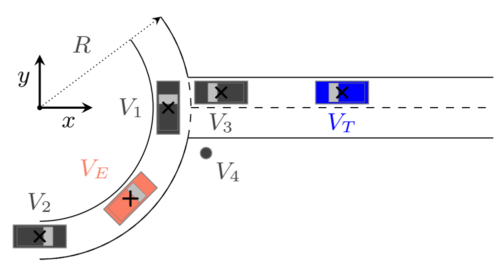

Fig. 2. Entering the roundabout. The black crosses symbolize the position vector $\boldsymbol{x}_i$ belonging to agent V_i.

The *rounD* dataset is another naturalistic dataset covering the scenario of roundabouts. There, V_T wants to enter the roundabout, either in front of or behind the ego vehicle V_E already inside the roundabout. Further agents are V_1 and V_2 driving respectively in front of or behind V_E inside the roundabout. V_3 might enter the roundabout directly in front of V_T, while V_4 is the pedestrian most likely to interact with V_4 (Fig. 2).

The following conditions

$$\begin{aligned}
C_S(t) &: y_1(t) = 5\,\mathrm{m} \wedge \dot{y}_1(t) > 0 \wedge \alpha_1(t) \in [0, 0.5\pi]\\
C_C(t) &: \alpha_E(t) = 0\\
C_A(t) &: r_T(t) = R \wedge \dot{r}_T(t) < 0
\end{aligned}$$

are now used for determining the characteristic time-points, with

$$t_{\widehat{C}}(t) = \frac{D_C(t)}{\max\left\{-\dot{D}_C(t), 0\right\}} + t$$
$$t_{\text{brake}}(t) = \frac{1}{2}\frac{\max\left\{-\dot{D}_C(t), 0\right\}}{a_{\text{brake}}}$$
$$\Delta t = 2.8\,\text{s}\,,$$

where

$$D_C(t) = \min\left\{R - r_E(t), 0\right\} \operatorname{sgn}\left(\alpha_E(t)\right) - R\alpha(t)$$
$$r(t)\exp\left(i\alpha(t)\right) = x(t) + iy(t)\,, \quad (\alpha \in [-\pi, \pi])\,.$$

C. Left turns

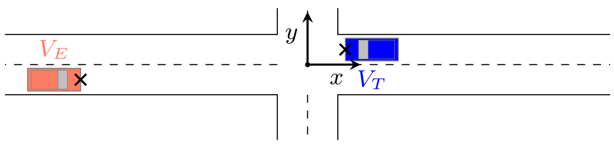

Fig. 3. The left turn. The black crosses symbolize the position vector $\boldsymbol{x}_i$ belonging to agent V_i.

The *L-GAP* dataset covers left turns at intersections, where the target vehicle V_T turns left across the planned trajectory of the ego vehicle V_E, which wants to drive straight ahead across the intersection (Fig. 3). Here, V_E appears when V_T has slowed down sufficiently during its approach to the intersection (t_S is set at this point). Therefore, this might not be the most challenging dataset, as then the onset of movement can be a obvious indicator of V_T intending to accept the gap. Furthermore, as there is no position data for V_E before t_S, adjustments are needed ($t_0 = t_S + (n_I - 1)\delta t$) to permit prediction at the opening of the gap. However, t_0 is often delayed enough so that the onset of movement is included in the input data provided to the model.

Nonetheless, the following conditions

$$C_C(t) : x_E(t) = -3\,\text{m}$$
$$C_A(t) : y_T(t) = 0\,\text{m} \wedge \dot{y}_T(t) < 0$$

are now used for determining the characteristic time-points, with

$$t_{\widehat{C}}(t) = -\frac{x_E(t) + 3\,\text{m}}{\max\left\{\dot{x}_E(t), 0\right\}} + t$$
$$t_{\text{brake}}(t) = \frac{1}{2}\frac{\max\left\{\dot{x}_E(t), 0\right\}}{a_{\text{brake}}}$$
$$\Delta t = 3.5\,\text{s}\,.$$

II. Splitting methods

A. Random splitting

The splitting is done separately for samples with accepted and rejected gaps so that the bias toward one of these groups should be roughly equal in both the training and testing set.

B. Extreme splitting

Here, the goal is to select those cases as testing samples, where the decision by the human actor V_T is most unintuitive. As for the previous method, the splitting is done separately for accepted and rejected gaps.

A rejected gap is considered more unintuitive if the available gap is larger. Accordingly, the samples with the highest values of $t_C - t_0$ become part of the test set. Meanwhile, accepting a gap is deemed more unintuitive for smaller gaps. Thus, those accepted gaps where $t_{\widehat{C}}(t_A) - t_A$ is lowest are used for testing.

III. Models

A. Trajectron++

The *Trajectron++* model uses many different neural network, such as long-short-term memories (LSTM) and convolutional neural networks (CNN). It requires velocity and acceleration data as inputs, necessitating some preprocessing of the given position data $\boldsymbol{X}_{\boldsymbol{T}_I}$ and $\boldsymbol{T}_I$. For the hyperparameters, the values used by the authors when applying the model to the *nuScenes* dataset [2] are selected [3], as this dataset also mainly includes motor vehicles. Furthermore, an attention radius between vehicles of $150\,\text{m}$ was chosen. No other changes have been made to the model.

B. AgentFormer

The *AgentFormer* model is a trajectory prediction model based on the concept of transformers (a technique proposed by Vaswani et al. [4]). Compared to other deep learning trajectory prediction models, which encode behavior over time and interactions between agents separately, this model processes all information simultaneously. For the hyperparameters, like for the previous model, the ones used by the authors when applying the model to the *nuScenses* dataset are selected [5].

C. Logistic Regression

While normally specifically selected properties are used as inputs of a logistic regression model, the positional data in $\boldsymbol{X}_I$ is given here. This results in an input with a dimensionality of $2n_I|\boldsymbol{V}|$. Although this is not done normally in the literature, where velocity data is common as well, this is no contradiction (as long as $n_I \geq 2$) since, during training, the model can extract the velocity data from the difference in positions at different time-steps:

$$a_x x_1 + b_x x_2 = a x_1 + b v_1$$
$$v_1 = \frac{x_2 - x_1}{\delta t}\,, \; a = a_x + b_x\,, \; b = b_x \delta t\,. \tag{4}$$

The same can be said for both relative distances and velocities and even accelerations (for the latter as long as $n_I > 2$). Due to the convexity of the optimized cost function during the model training, it will ultimately arrive at the same result, despite the linear transformation of the inputs.

D. Random forest

As defined by Nagalla et al. [6], random forests can have two hyperparameters, the number of bagged trees and the number of features considered for splits. Those are chosen in this implementation by a grid search with ten-fold cross-validation.

E. Deep Belief Network

For the deep belief network, the approach for determining the network structure (i.e., number of layers and nodes) is taken from Xie et al. [7]. Compared to that work, which relied on relative distances and velocities as inputs, trajectories are here used unaltered. Still, as the first step in the model is a multiplication of a weight matrix to the input, the lack of velocities should not be a problem either, as a linear transformation can be counteracted (see above). However, as the cost function $\mathcal{L}$ of a deep neural network is not convex, this cannot be guaranteed.

F. Meta-heuristic model

The meta-heuristic model is created by combining the predictions of multiple other models. Here, such a model is created by combing the predictions a_p of the other models on a sample with the sample's original input $\boldsymbol{X}_{\boldsymbol{T}_I}$. A logistic regression model—which according to Khelfa et al. [8] is the most promising approach—is then trained on these expanded inputs.

IV. Metrics

A. Accuracy

$$F = \frac{1}{N_A + N_{\overline{A}}} \left|\left\{i \,|\, a_i = 0 \veebar a_{\text{pred},i} > \tau \right\}\right| \tag{5}$$

Here, τ is the decision threshold chosen to maximize the final value of F. For a random binary predictor, this threshold would be either 0 or 1 (depending on the bias in the dataset), resulting in

$$F_r = \frac{\max\left\{N_A, N_{\overline{A}}\right\}}{N_A + N_{\overline{A}}} \,. \tag{6}$$

B. AUC

$$F = \frac{1}{N_A N_{\overline{A}}} \left(\left(\sum_i a_i r_i\right) - \frac{N_A\left(N_A + 1\right)}{2}\right) \tag{7}$$

Here, r_i is the position of sample i according to its value $a_{\text{pred},i}$ ($r_i = 1$ for the lowest value of a_{pred} and $r_i = N_A + N_{\overline{A}}$ for the highest value). Here, one gets $F_r = 0.5$.

C. Average displacement error

$$\begin{aligned} F =& \frac{1}{\left(N_A + N_{\overline{A}}\right)\lceil n_p \beta \rceil} \sum_i \sum_{p=1}^{\lceil n_p \beta \rceil} D_{i,p} \\ D_{i,p} =& \sum_{t \in \boldsymbol{T}_{O,i}} \frac{1}{n_{O,i}} \left\| \boldsymbol{x}_{T,i,p}(t) - \boldsymbol{x}_{T,i}(t) \right\| \end{aligned} \tag{8}$$

with

$$D_{i,p} \leq D_{i,p+1} \;\forall\, p \in \{1, \ldots, n_p - 1\} \,.$$

D. True negative rate for perfect recall

$$F = \frac{\left|\left\{i \,|\, a_i = 0 \wedge a_{\text{pred},i} < \min\left\{a_{\text{pred},i} \,|\, a_i = 1\right\}\right\}\right|}{n_{\overline{A}}} \tag{9}$$

For a uniformly random predictor, one would get

$$F_r = \min\left\{a_{\text{pred},i} \,|\, a_i = 1\right\} = \frac{1}{N_A + 1} \,. \tag{10}$$

References

[1] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems," in *IEEE Int. Conf. on Intell. Transp. Syst. (ITSC)*, pp. 2118–2125, Nov. 2018.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *IEEE/CVF Conf. on Comput. Vis. Pattern Recognit.*, pp. 11621–11631, 2020.

[3] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data," Aug. 2022.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[5] Y. Yuan, X. Weng, Y. Ou, and M. Kitani, "AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting," Aug. 2022.

[6] R. Nagalla, P. Pothuganti, and D. S. Pawar, "Analyzing Gap Acceptance Behavior at Unsignalized Intersections Using Support Vector Machines, Decision Tree and Random Forests," *Procedia Comput. Sci.*, vol. 109, pp. 474–481, Jan. 2017.

[7] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He, "A data-driven lane-changing model based on deep learning," *Transp. Research Part C: Emerg. Technol.*, vol. 106, pp. 41–60, Sept. 2019.

[8] B. Khelfa, I. Ba, and A. Tordeux, "Predicting highway lane-changing maneuvers: A benchmark analysis of machine and ensemble learning algorithms," Apr. 2022. arXiv:2204.10807[physics].