

Benchmarking Behavior Prediction Models in Gap Acceptance Scenarios

Schumann, Julian F.; Kober, Jens; Zgonnikov, Arkady

DOI

[10.1109/TIV.2023.3244280](https://doi.org/10.1109/TIV.2023.3244280)

Publication date

2023

Document Version

Final published version

Published in

IEEE Transactions on Intelligent Vehicles

Citation (APA)

Schumann, J. F., Kober, J., & Zgonnikov, A. (2023). Benchmarking Behavior Prediction Models in Gap Acceptance Scenarios. *IEEE Transactions on Intelligent Vehicles*, 8(3), 2580-2591. <https://doi.org/10.1109/TIV.2023.3244280>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Benchmarking Behavior Prediction Models in Gap Acceptance Scenarios

Julian F. Schumann , Jens Kober , and Arkady Zgonnikov 

Abstract—Autonomous vehicles currently suffer from a time-inefficient driving style caused by uncertainty about human behavior in traffic interactions. Accurate and reliable prediction models enabling more efficient trajectory planning could make autonomous vehicles more assertive in such interactions. However, the evaluation of such models is commonly oversimplistic, ignoring the asymmetric importance of prediction errors and the heterogeneity of the datasets used for testing. We examine the potential of recasting interactions between vehicles as gap acceptance scenarios and evaluating models in this structured environment. To that end, we develop a framework aiming to facilitate the evaluation of any model, by any metric, and in any scenario. We then apply this framework to state-of-the-art prediction models, which all show themselves to be unreliable in the most safety-critical situations.

Index Terms—Autonomous vehicles, gap acceptance, behavior prediction, benchmark.

I. INTRODUCTION

SUCCESSFULLY implementing autonomous driving is one of the key technical challenges faced by the automotive industry as well as large parts of the research community, with tens of billions of dollars invested in recent years towards this goal [1]. The provision of those funds is motivated by several benefits promised by this technology. The foremost of these is safer driving, expressed by a significant decrease in accidents and, correspondingly, a reduction of bodily harm and financial losses. Additional advantages are also expected, such as more accessible mobility for people unable to drive or an easing of road congestion and traffic [2], [3], [4].

But despite all these investments, autonomous vehicles still suffer from many problems preventing widespread use [5], [6]. One such problem is their timidity in interactions with human traffic participants, caused by the uncertainty about the future behavior of those human agents. This uncertainty can prevent the autonomous vehicle from taking the most time-efficient actions if the resulting probability of a crash or near-crash is too high, resulting in the cautious driving style observed. Paradoxically,

this can also be a safety risk, as such caution by an autonomous vehicle is often not expected by the surrounding humans, which can result in accidents such as being rear-ended [5], [7].

To reduce this uncertainty and to allow for a more efficient driving style without compromising on safety requirements, behavior prediction models can be used [8], [9], which project the future position of traffic participants. Those can range from models able to deal with any kind of traffic participant [10], [11], [12] to others focused on predicting the behavior of a specific kind of participant, such as cars [13], [14], [15], [16], [17], [18], [19] or pedestrians [20], [21], [22], [23], [24].

However, the utility of those models—primarily designed to minimize the necessary trade-off between safety and efficiency in trajectory planning—is questionable, as the common methods for their evaluation diverge from the models' purpose. First, most common metrics for evaluating prediction models, such as the final or average displacement error, ignore that the consequences of a false prediction are inherently asymmetric [25], [26]. For example, on a highway, wrong longitudinal predictions are far less dangerous than wrong lateral predictions, which might result in an autonomous vehicle reacting to a lane change too late. Similarly, such metrics also lack the ability to evaluate how good models are at capturing distinct human behaviors [27], [28]. Second, the common approach of randomly selecting test cases from datasets [10], [12] is problematic due to the heterogeneity of those datasets, which typically include samples that can vary widely in their importance and difficulty. Such samples can range from a single vehicle following a lane to complex space-sharing conflicts with multiple agents at unsignalized intersections, where the behavior of human agents is often multi-modal and can change rapidly. Rare edge cases, where some traffic participants are very aggressive or even violate traffic rules and accidents are far more likely [29], [30], are also possible. But with randomly selected test cases, potentially poor performance in the most important situations can be compensated by good performance in less important but more numerous ones. For these reasons, a model that is unreliable in actual safety-critical situation might still appear promising, making the whole evaluation meaningless and hampering further progress.

One possible approach to overcome these issues is including a path planning algorithm in evaluations, as suggested by Ivanovic and Pavone [25]. However, this adds further computational loads to an evaluation and only addresses the symmetry of common metrics, neglecting the varying difficulty and importance between testing samples. To cover both these problems, we suggest instead narrowing the evaluation to the most critical situations. In

Manuscript received 2 January 2023; revised 19 January 2023; accepted 6 February 2023. Date of publication 13 February 2023; date of current version 27 April 2023. (Corresponding author: Julian F. Schumann.)

The authors are with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, ZH, The Netherlands (e-mail: J.F.Schumann@tudelft.nl; j.kober@tudelft.nl; A.Zgonnikov@tudelft.nl).

The source code, trained models, and data can be found online at public Github repository (<https://github.com/julianschumann/Framework-for-benchmarking-gap-acceptance>).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TIV.2023.3244280>.

Digital Object Identifier 10.1109/TIV.2023.3244280

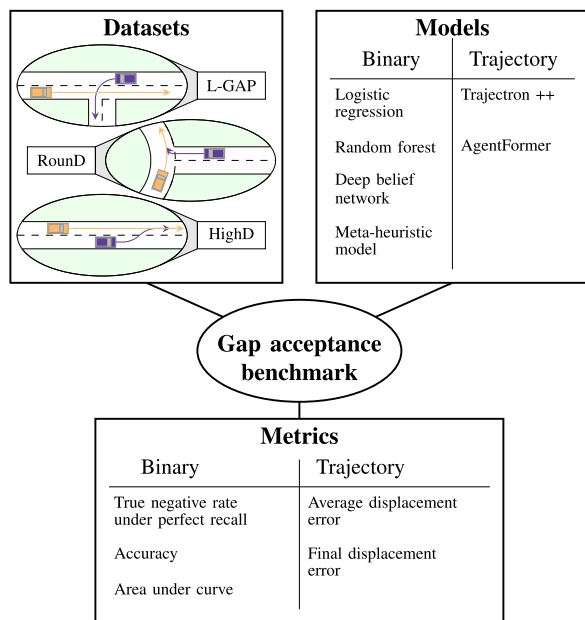


Fig. 1. The proposed framework allows researchers to evaluate the performance of various prediction models for human behavior according to several metrics on different datasets that include gap acceptance scenarios. In this work, three datasets, six models, and four metrics (dealing both with binary and trajectory predictions) are implemented.

particular, we focus the evaluation of behavior prediction models on *gap acceptance* scenarios, a concept that encompasses most of the safety critical interactions between autonomous vehicles and humans [31]. In a gap acceptance scenario, an autonomous vehicle follows a particular trajectory over which a second traffic participant (e.g., a pedestrian or another vehicle) can move either in front of or behind the autonomous vehicle. Here, the first option (i.e., the human accepting the gap) would require the autonomous vehicle to potentially alter its trajectory planning, while the latter one of rejecting the gap would not. Due to the narrow focus, estimating the importance and difficulty of a particular situation can become much more straightforward. Additionally, as the human has only two options to decide between, such gap acceptance scenarios allow the usage of simple binary prediction models to estimate if the human behavior requires an adjustment of trajectory planning.

Many binary prediction models have been developed for gap acceptance scenarios. However, those mostly focus and are trained on a specific scenario, such as the street crossing behavior of pedestrians [32], [33], [34], [35], the crossing behavior of cars at intersections [36], [37], [38], [39], [40], or lane change decisions on high ways [41], [42], [43], [44], [45]. Additionally, the development of those models still suffers from similar problems as the trajectory prediction models, such as the anisotropy of common metrics like accuracy. Likewise, a random selection of test cases [33], [38] and neglect of the varying importance of different samples are also common. Additionally, in contrast to trajectory prediction models, which are commonly compared to each other on accepted benchmarks (such as on the ETH dataset [10], [11], [12] when predicting pedestrian crowds), an equivalent benchmark does not exist for binary prediction

models [46]. Instead, those models are mostly trained and tested on datasets exclusive to the respective work and are—if at all—only compared against a small number of other selected models [32], [33], [38], [39], [40], [47], [48], [49], [50], [51].

Our goal in this work is to overcome these limitations of the current literature on both binary and trajectory prediction models and enable a meaningful evaluation of these models in gap acceptance scenarios. Such an evaluation cannot only make the development of trajectory prediction models more goal-oriented but also help determine to what extent the inclusion of specialized binary prediction models can improve the performance and reliability of general trajectory prediction models. To that end, this article makes three main contributions:

- We develop a formal description of the gap acceptance process that applies to all possible gap acceptance scenarios. This description includes a detailed timeline of gap acceptance (Section II), which serves as a foundation for methods to estimate the criticality concerning the safety of each sample, which is a fundamental requirement for selecting meaningful test cases.
- We devise a framework for evaluating behavior prediction models in gap acceptance scenarios. This novel framework allows the integration of varied gap acceptance datasets, models, and evaluation metrics (Section III). It is inspired by similar works by Müller et al. for computer-based image retrieval algorithms [52], by Zaffar et al. in the field of visual place recognition [53], or Cao et al. for evaluating robustness to adversarial attacks of trajectory prediction models [54]. This approach would allow one to test and compare models in a chosen environment more easily, no matter if they make binary or trajectory predictions. For example, the proposed framework enables evaluating trajectory prediction models using binary metrics in gap acceptance scenarios, something that has not been previously investigated. Simultaneously, the framework allows precise control over the splitting of the data into training and testing samples to comprehensively evaluate the models' reliability in the most difficult gap acceptance situations.
- Using the proposed framework, we compare six prediction models in their performance on three gap acceptance datasets (Section IV and Fig. 1). First, this demonstrates the general viability of our proposed method, by including models making and metrics evaluating both binary and trajectory predictions. Second, it allows us to test the reliability of those models specifically in safety-critical edge cases, which is currently missing from the literature. Third, it allows the testing of the hypothesis that including dedicated binary models for gap acceptance problems could improve the performance of state-of-the-art trajectory prediction models. Lastly, this gives researchers easy access to an already implemented baseline to compare their own behavior prediction models to.

II. DEFINING GAP ACCEPTANCE

To estimate the difficulty and control the importance of prediction tasks over disparate datasets, a coherent formal definition

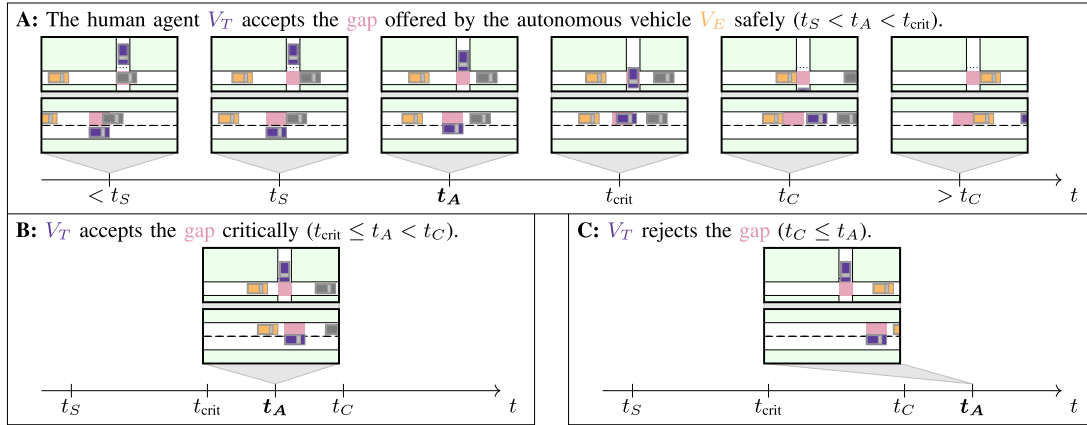


Fig. 2. The characteristic time-points of the gap acceptance process—defined by the relation of the agents to the contested space (in purple)—of two different examples of gap acceptance, intersection crossing (upper panels) and lane changing (lower panels). In both examples, the autonomous vehicle V_E (in red) offers a gap to the human-driven vehicle V_T (in blue). In total, three cases are possible (A – C), depending on t_A , i.e., the time the target vehicle enters the contested space. In B, the accepted gap decision by the human is considered to be unsafe, as V_E cannot guarantee the avoidance of a crash, having potentially not enough time for braking. Meanwhile, in C, it might be possible that V_T crashes into V_E .

of gap acceptance scenarios is needed. Here we propose such a definition.

In a gap acceptance scenario, an autonomous vehicle V_E —also referred to as the ego-vehicle—plans to follow along a certain trajectory P_E along which it has the right of way. This trajectory overlaps with the trajectory P_T of another, human-controlled vehicle V_T (also named target vehicle). Such an overlap might, for example, happen at unsignalized intersections, where the agents move along crossing streets or on highways, where V_T wants to merge into the faster lane along which V_E is driving. In such situations, V_T can decide to move onto P_E either in front of or behind V_E , i.e., to accept or reject the gap offered by V_E . We assume that V_E has the right of way along P_E , as otherwise, traffic rules would obligate it to preemptively yield.

Under these conditions, a gap acceptance scenario is characterized by the spatiotemporal relation between the agents towards the so-called contested space [31]. There, the trajectories P_E and P_T would start to overlap, making this the location of a potential collision. An example is the overlap of two crossing lanes at an intersection. However, in specific scenarios (such as changing lanes on highways), the exact location of the meeting point of P_E and P_T can be at the discretion of the human agent V_T and therefore be unknown before the actual decision. In such cases, we then place the contested space under the assumption that V_T would decide to accept the gap immediately. For example, in the scenario of highway lane changes, the contested space would therefore move in parallel to V_T , only stopping to move once V_T starts to enter the lane of V_E .

The following time points then characterize the gap acceptance process (illustrated together with the contested space in Fig. 2):

t_S : At the starting time t_S , there is no longer any other vehicle along P_E in between V_E and the contested space. This is primarily the case when the vehicle preceding V_E leaves the contested space, but other options are imaginable, like the vehicle in front of V_E leaving P_E .

t_C : At t_C , V_E starts to enter the contested space, closing the gap.

$t_{\underline{C}}(t)$: A prediction of t_C by the ego vehicle, made at t , needed to allow gap size estimations during online applications. While this is scenario-dependent, the following condition has to be satisfied so that an open gap can still be characterized as such, even if V_E is moving away from the contested space:

$$\text{sgn}(t_{\underline{C}}(t) - t) = \text{sgn}(t_C - t) .$$

t_{crit} : The last time V_E can safely prevent a collision even in the case of malicious behavior by V_T ; e.g., at this point, a safe braking process could bring V_E to a stop before the intersection. t_{crit} can be formalized in the following condition:

$$\Delta t_D(t) = t_{\underline{C}}(t) - t - t_{brake}(t) = 0 \quad (1)$$

Here, the required braking time t_{brake} is not based on the maximum deceleration V_E is technically capable of, but instead, one that is considered safe. The time point t_{crit} is also the last time a prediction can be considered useful for further trajectory planning.

t_A : At t_A , V_T enters the contested space, potentially accepting the gap.

We count V_T as rejecting the gap if V_E is allowed to move first onto the contested space, i.e., if $t_C \leq t_A$. If this is not the case and the human moves first ($t_A < t_C$), the gap is considered accepted.

III. FRAMEWORK FOR BENCHMARKING GAP ACCEPTANCE MODELS

After defining the fundamental characteristics of a gap acceptance scenario, we will use this groundwork to build a framework for benchmarking gap acceptance models. This framework should allow for the performance assessment of a prediction

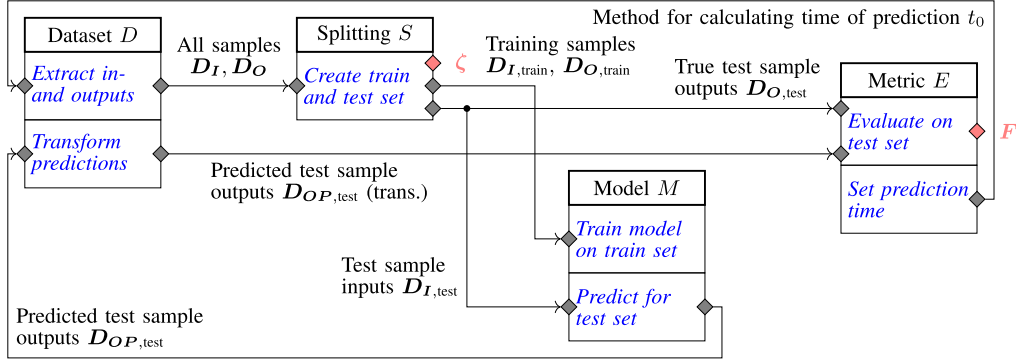


Fig. 3. Functionalities of the proposed framework. To evaluate a model M on a dataset D with the metric E and splitting method S , a method for determining the prediction time t_0 has to be chosen first (see Section III-A for a detailed description). This method is used to extract input and output trajectories (D_I and D_O respectively) from the dataset D (III-B). Those samples are split in training and testing set using the splitting method S (III-C), with the training one ($D_{I,\text{train}}$, $D_{O,\text{train}}$) used to train a model M (III-D). Subsequently, predictions $D_{OP,\text{test}}$ are made for the test samples $D_{I,\text{test}}$ with the trained model (III-E). It might be necessary to transform these predictions into another form (III-F), before the metric E compares them to the true outputs $D_{O,\text{test}}$ (III-G). These steps produce two outcomes (red diamonds): the similarity ζ between training and test data, provided by S , and the model performance F according to metric E .

model M on a dataset D according to evaluation metric E . The following requirements need to be met for such an assessment to be both meaningful and possible for as many of the aforementioned modules as possible:

- R 1** The time point t_0 of a prediction must be controllable, as it influences not only the difficulty of the prediction but also its importance due to changing consequences of a false prediction.
- R 2** To evaluate models in critical situations, the framework should allow control over splitting all available samples into training and testing sets.
- R 3** Models producing (as well as metrics evaluating) for example binary or trajectory predictions should fit into the framework. Therefore, the framework should allow transformations between those forms of model output.

Considering these requirements, seven functionalities will constitute the proposed framework. Fig. 3 illustrates these functionalities and their mutual dependencies. These functionalities are grouped in four modules (dataset D , splitting method S , model M , and Metric E); in this section we describe them in the order in which they are employed in the process of a single evaluation.

A. Setting the Prediction Time — Metric E

To satisfy requirement **R 1**, this functionality enables the selection of the time-point t_0 at which the prediction has to be made. As the prediction time influences both the importance of such predictions and the meaningfulness of different metrics (Appendix A), this functionality is attached to the metric module.

Currently, three methods are implemented into the framework to determine t_0 :

- *Prediction at the initial opening of the gap:* $t_0 = t_S$. The prediction is made when the gap first appears, and this is the baseline most commonly used in the literature [38], [42], [47].

- *Prediction at gaps with fixed size:* $t_0 = \min\{t \mid t_{\underline{C}}(t) - t = \Delta t\}$. The prediction is made when the gap offered has a uniform duration Δt , which should make every prediction equally difficult due to a similar prediction horizon.
- *Last useful prediction for critical gaps:* $t_0 = t_{\text{crit}} - t_e$. The prediction is made at the last point in time when it would still be useful, with t_e being used to allow time for calculations.

Here, t_0 has to be calculated without hindsight knowledge for online predictions at a time when t_C or t_A are not known. A discussion on the impact of the different approaches on the resulting datasets can be found in Appendix A.

B. Extracting Input and Output — Dataset D

Next, the input and output data for each sample are extracted from a given trajectory \mathbf{X}_T , which includes positions at different time points T from different actors $\mathbf{V} = \{V_E, V_T, V_1, \dots\}$:

$$\begin{aligned} \mathbf{X}_T &= \{\mathbf{x}(t) \mid t \in T\}, \text{ where } I_T = [\max T, \min T] \\ \mathbf{x}(t) &= \{\mathbf{x}_i(t) \mid V_i \in \mathbf{V}\} \\ \mathbf{x}_i(t) &= (x_i(t), y_i(t)) \in \mathbb{R}^2 \end{aligned}$$

This functionality, requiring access to the raw data from the scenario and thus being part of the dataset module, consists of ten consecutive steps:

- 1) $t_{\underline{C}}$ and t_{brake} are estimated at every time point in T , which is scenario-dependent.
- 2) The characteristic time points t_S , t_A and t_C are determined. Using the scenario-specific conditions C_S , C_C , and C_A respectively (C_S is true if the gap is offered, C_C and C_A are true if respectively the ego vehicle V_E and target vehicle V_T are inside the contested space), one then has to find the specific times $T_{C_i} = \{t \mid C_i(t) \forall t \in I_T\}$ at which those conditions are true (e.g., T_{C_A} is the time during which V_T is inside the contested space); from

this, the characteristic time points are extracted:

$$\begin{aligned} t_S &= T_S(\mathbf{X}_T) = \begin{cases} \min \mathbf{T} & \mathbf{T}_{C_S} = \emptyset \\ \max \mathbf{T}_{C_S} & \text{else} \end{cases} \\ t_C &= T_C(\mathbf{X}_T) = \begin{cases} t_{\underline{C}}(\max \mathbf{T}) & \mathbf{T}_{C_C} = \emptyset \\ \min \mathbf{T}_{C_C} & \text{else} \end{cases} \\ t_A &= T_A(\mathbf{X}_T) = \begin{cases} \max \mathbf{T} + t_\epsilon & \mathbf{T}_{C_A} = \emptyset \\ \min \mathbf{T}_{C_A} & \text{else} \end{cases} \end{aligned} \quad (2)$$

Here, a sample is excluded from the dataset, if no decision can be observed (i.e., if $\mathbf{T}_{C_C} = \emptyset \wedge \mathbf{T}_{C_A} = \emptyset$).

- 3) The binary decision a , with $a = 1$ for accepted gaps ($t_A < t_C$) and $a = 0$ for rejected gaps, is extracted.
- 4) t_{crit} is extracted next, with

$$t_{\text{crit}} = \begin{cases} t_S & \Delta t_D(t_S) \leq 0 \\ t_A + t_\epsilon & \min\{\Delta t_D(t) \mid t_S \leq t < t_A\} > 0, \\ t_D & \text{else} \end{cases}$$

where

$$t_D = \min\{t \mid t > t_S \wedge \Delta t_D(t) = 0\}$$

satisfies both requirements in (1).

- 5) The time of prediction t_0 is calculated accordingly to the method chosen previously (III-A). Only samples that meet the condition

$$t_S \leq t_0 < \min\{t_A, t_{\text{crit}}\} \quad (3)$$

are included in the final dataset, to ensure that gaps are already offered, V_T has not made a decision yet, and that the prediction is still useful.

- 6) The number of input time-steps n_I and the time-step size δt are chosen.
- 7) One also has to determine the number of output time steps n_O :

$$n_O = \left\lceil \frac{t_C - t_0}{\delta t} \right\rceil \quad (4)$$

The resulting prediction horizon $n_O \delta t$ is therefore large enough to see the outcome of the gap acceptance scenario, i.e. the acceptance or closing of the gap.

- 8) Based on t_0 , n_I , n_O , and δt , the time-steps for input and output data are selected, named \mathbf{T}_I and \mathbf{T}_O respectively:

$$\begin{aligned} \mathbf{T}_I &= \{t_0 + i\delta t \mid i \in \{-n_I + 1, \dots, 0\}\} \\ \mathbf{T}_O &= \{t_0 + i\delta t \mid i \in \{1, \dots, n_O\}\} \end{aligned} \quad (5)$$

- 9) For those time-steps, the input trajectories $\mathbf{X}_{\mathbf{T}_I}$ and output trajectories $\mathbf{X}_{\mathbf{T}_O}$ are extracted from \mathbf{X}_T , using interpolation if necessary.
- 10) Certain domain information k is collected, for instance, the location at which the trajectories \mathbf{X}_T were collected or the test subjects involved in gathering the data.

The input data \mathbf{D}_I then includes from each sample the input trajectory $\mathbf{X}_{\mathbf{T}_I}$ and the corresponding time-steps \mathbf{T}_I . Meanwhile, the output data \mathbf{D}_O takes the output trajectory $\mathbf{X}_{\mathbf{T}_O}$ and

the corresponding time-steps \mathbf{T}_O , as well as the binary decision a , the time of accepting the gap t_A , and the domain information k from each sample.

C. Creating Training and Testing Set — Splitting Method S

To fulfill requirement **R 2**, the splitting method S , separating the given samples created in the previous step (III-B) into training and testing sets, is a crucial part of the framework. As this functionality should be independent of the scenario, it is part of the separate splitting module. Examples for this range from random splitting to methods taking into account all the information in \mathbf{D}_I and \mathbf{D}_O .

Besides the potential similarity measure ζ of training samples to the training set, this functionality creates the training data $\mathbf{D}_{I,\text{train}}$ and $\mathbf{D}_{O,\text{train}}$ as well as the test data $\mathbf{D}_{I,\text{test}}$ and $\mathbf{D}_{O,\text{test}}$.

D. Training the Model on the Training Set — Model M

After splitting the samples into training and testing sets (III-C), the model has to be trained on the training set ($\mathbf{D}_{I,\text{train}}$, $\mathbf{D}_{O,\text{train}}$), which is one of the functionalities of the model module that has to be individually implemented for each model. If the model, for instance, requires input velocities, extracting those from the given position data $\mathbf{X}_{\mathbf{T}_I}$ and $\mathbf{X}_{\mathbf{T}_O}$ is done here.

E. Making the Predictions for the Testing Set — Model M

For every sample from the input testing set $\mathbf{D}_{I,\text{test}}$, a prediction \mathbf{d}_{pred} is made by the model trained previously (III-D), with all \mathbf{d}_{pred} constituting the set of predictions $\mathbf{D}_{OP,\text{test}}$. As such predictions rely on a trained model, this functionality is also part of the model module. Depending on this model, each prediction \mathbf{d}_{pred} might take different forms. Three different forms of stochastic predictions are implemented into the framework.

- Binary prediction: $\mathbf{d}_{\text{pred}} = a_{\text{pred}}$, i.e., only the probability $a_{\text{pred}} \in [0, 1]$ of V_T accepting the offered gap is predicted.
- Timing prediction: $\mathbf{d}_{\text{pred}} = \{a_{\text{pred}}, t_{A,\text{pred}}\}$, i.e., not only a_{pred} is predicted but also the time $t_{A,\text{pred}}$ at which the gap acceptance might take place (Appendix A).
- Trajectory prediction: $\mathbf{d}_{\text{pred}} = \mathbf{X}_{\mathbf{T}_O,\text{pred}}$, i.e., the full trajectory of V_T is predicted. This prediction consists of n_p trajectories $\mathbf{X}_{\mathbf{T}_O,p}$ —all equally likely—to represent probabilistic outputs:

$$\mathbf{X}_{\mathbf{T}_O,\text{pred}} = \{\mathbf{X}_{\mathbf{T}_O,1}, \dots, \mathbf{X}_{\mathbf{T}_O,n_p}\}$$

F. Transforming the Predictions — Dataset D

To fulfill requirement **R 3**, we then must be able to transform a prediction \mathbf{d}_{pred} from the previous step (III-E) to another prediction form if necessary. This functionality is a part of a specific dataset, as this requires the context information to, for example, classify different trajectories as accepted or rejected gap.

To facilitate those transformations, three different functions T_i (exact implementation in Appendix A) are needed, as can be seen in Table I:

TABLE I
TRANSFORMATION BETWEEN THE THREE POSSIBLE PREDICTION TYPES d_{pred} IMPLEMENTED INTO THE PROPOSED FRAMEWORK

Input \Rightarrow Output	a_{pred}	$\{a_{\text{pred}}, t_{A,\text{pred}}\}$	$\mathbf{X}_{T_O,\text{pred}}$
Binary prediction a_{pred}	—	$\{a_{\text{pred}}, T_3(a_{\text{pred}})\}$	$T_2(\{a_{\text{pred}}, T_3(a_{\text{pred}})\})$
Timing prediction $\{a_{\text{pred}}, t_{A,\text{pred}}\}$	—	—	$T_2(\{a_{\text{pred}}, t_{A,\text{pred}}\})$
Trajectory prediction $\mathbf{X}_{T_O,\text{pred}}$	in $T_1(\mathbf{X}_{T_O,\text{pred}})$	$T_1(\mathbf{X}_{T_O,\text{pred}})$	—

Using the three implemented functions T_i .

TABLE II
THE NUMBER OF ACCEPTED GAPS N_A AND REJECTED GAPS $N_{\neg A}$ IN THE IMPLEMENTED DATASETS

Dataset	Initial gaps at their opening	Gaps with fixed size	Critical gaps
highD (Lane changes)	1406 – 7026 (8.9 s)	461 – 1568 (11.8 s)	0 – 7025 (0.8 s)
highD (Lane changes - restricted)	1406 – 1001 (12.9 s)	392 – 241 (8.7 s)	0 – 1000 (0.7 s)
roundD (Roundabout)	662 – 917 (2.5 s)	168 – 168 (2.9 s)	33 – 913 (1.0 s)
L-GAP (Left turns)	703 – 724 (4.6 s)	496 – 572 (3.5 s)	369 – 723 (2.3 s)

In the form: $N_A - N_{\neg A}$ (median $t_c(t_0)$). The numbers are depicted for three different methods of choosing t_0 .

- T_1 : Takes the trajectory prediction $\mathbf{X}_{T_O,\text{pred}}$ and then provides $\{a_{\text{pred}}, t_{A,\text{pred}}\}$. This is similar to the extraction of the time points in Section III-B.
- T_2 : Takes the prediction $\{a_{\text{pred}}, t_{A,\text{pred}}\}$ and then provides the trajectory prediction $\mathbf{X}_{T_O,\text{pred}}$, consisting of n_p trajectories from the predictions of two conditional trajectory prediction models trained only on accepted and rejected gaps respectively. These are selected so that $T_1(\mathbf{X}_{T_O,\text{pred}})$ results in the original inputs.
- T_3 : Takes a binary prediction a_{pred} and provides the predicted time of accepting the gap $t_{A,\text{pred}}$, by extracting it from the prediction of a trajectory prediction model trained only on accepted gaps.

G. Evaluating the Predictions — Metric E

This functionality—the main part of the metric module—implements the performance evaluation, comparing the actual outputs $D_{O,\text{test}}$ (from III-C) with the predicted outputs $D_{OP,\text{test}}$ (III-F). It returns either a combined value F or instead a separate value for each sample $d_{\text{pred}} \in D_{OP,\text{test}}$, resulting in the output F .

IV. BENCHMARK IMPLEMENTATION

We implemented the framework described above by linking together several datasets, models, splitting methods, and metrics. These were chosen not to comprehensively cover all possible gap acceptance scenarios and prediction models but to demonstrate the flexibility and utility of the proposed framework. Still, our implementation can already serve as a benchmark for new prediction models. This section only presents an overview of the implementation, with full technical specification provided (<https://github.com/julianschumann/Framework-for-benchmarking-gap-acceptance/blob/main/Framework/Benchmark-Implementation.pdf>) in the supplementary materials.

A. Datasets

Different datasets are implemented into the framework (Table II), including data recorded on real roads as well as data from a driving simulator study. The naturalistic datasets used here are captured by drones and distinguished by accurate position labeling. They cover lane changes on German highways (the *highD* dataset [55]) and roundabouts (the *roundD* dataset [56]). The L-GAP dataset covers left turns at unsignalized intersections through oncoming traffic recorded in a driving simulator [38]. It has been chosen due to the simplicity of its environment, contrasting the more complex scenarios in the naturalistic datasets.

1) *Lane Changes*: Here we focus on lane changes of the target vehicle V_T toward a faster lane to the left, along which the ego vehicle V_E driving there has the right of way. While it could be argued that predictions in such situations could be simply based on turn signals, one cannot rely on human drivers to correctly use these [50]. As a source of lane change data, we used two versions of the *highD* dataset, *full* and *restricted*.

The full *highD* dataset, not employing any filters, is heavily biased toward trajectories without a lane change. This is not a problem per se, but in such trajectories it is not known whether the target vehicle V_T even had an intention to change lanes (i.e., if there was a gap acceptance situation in the first place). For this reason, in addition to the full *highD* dataset, we added a restricted version of it which only included samples for which it can be inferred that the target vehicle V_T indeed considered a lane change. Criteria are either a lane change of V_T after V_E has passed or V_T braking to not collide with the preceding vehicle instead of changing lanes. Still, in both versions of *highD*, the gaps are always accepted with large safety margins (Table II).

2) *Roundabout*: In the *roundD* dataset, the target vehicle V_T has to enter a roundabout, which it can do in front of or behind the ego vehicle V_E already in the roundabout. As the trajectories are recorded in Germany, the ego vehicle inside the roundabout has the right of way. Compared to *highD*, this dataset is far

more balanced between accepted and rejected gaps, but still only includes few critically accepted gaps.

3) *Left Turns*: In the *L-GAP* dataset [38], the driver of the target vehicle V_T intends to turn left at an intersection. The driver had to decide whether to do this in front of or behind the ego vehicle V_E approaching the intersection from the opposite direction with the right of way. While the number of samples in this dataset is comparatively small, they are relatively balanced between accepted and rejected gaps. Also, they include many gaps accepted after t_{crit} (Table II). Nonetheless, as V_T starts in an idling position at some distance to the contested area, this might not be the most challenging dataset, as an onset of movement before t_0 in most cases is an apparent indicator of V_T intending to accept the gap.

B. Test-Train Splitting Methods

Two splitting methods are implemented, without a method for calculating the similarity measure ζ . Nonetheless, to enable at least a qualitative approximation of a model's robustness, the methods are designed to produce testing sets of varying difficulty for the prediction models.

The easier variant performs a stratified random splitting, while the second, more extreme method sorts the most unintuitive behavior of the target vehicle into the testing set (e.g. accepting a very small gap or rejecting a very large gap).

In both cases, the testing set includes 20% of the samples and the training set the remaining 80%.

C. Models

The benchmark includes two state-of-the-art trajectory prediction models

- *Trajectron++* (also referred to as *T+*), a deep-learning model mainly based on long-short-term memory cells [10].
- *AgentFormer (AF)*, a deep-learning model based on transformers [12]. Compared to *T+*, it has ten times more trainable parameters.

For the binary prediction models for gap acceptance, there is, as mentioned above, a lack of a common benchmark, making the models' selection more contentious. Four models have been selected nonetheless:

- Logistic regression (*LR*) is commonly used for predicting human gap acceptance decisions [34] and is therefore included as a simple baseline.
- Random forests (*RF*) have been shown to outperform other approaches such as logistic regression and standard decision trees in gap acceptance prediction [36].
- Deep belief networks (*DB*), also used previously to predict human gap acceptance decisions [42].
- A metaheuristic model based on combining all other five models above (*MH*); previously a similar approach for lane changes has been shown to outperform each of the models included in it [44].

The benchmark does not include any dedicated timing prediction models yet, as their primary representative, the drift-diffusion model [33], [38], can currently not be trained on datasets with a large number of unique samples in a reasonable

amount of time. Nonetheless, to allow for future expansion of the benchmark, the framework has been designed with such models in mind.

D. Evaluation Metrics

We have included several metrics that characterize models in terms of the quality of binary predictions (accept/reject gap) as well as full trajectory predictions. The following metrics are commonly used in the literature:

- *Accuracy*: This metric is a widespread method to evaluate the performance for binary prediction models [36], [42], [44]. However, accuracy is a symmetric metric, i.e. it is unable to differentiate between false negative and false positive predictions. Consequently, it is best used in cases where $t_0 \ll t_{\text{crit}}$, as the consequences of false predictions are not too different there (Appendix A).
- *AUC*: This metric for binary prediction models, the Area Under Curve of a receiver operating characteristics curve, addresses one central point of criticism of the accuracy metric, namely its sensitivity to biases in the testing set. Nonetheless, like the accuracy metric, it does not consider the potentially differing severity of false predictions. Hence, we only apply it to rate a prediction model's performance when $t_0 \ll t_{\text{crit}}$.
- *ADE $_{\beta}$* and *FDE $_{\beta}$* : These metrics, the Average or Final Displacement Error of the $n_p\beta$ least erroneous predicted trajectories, are commonly applied to trajectory predictions [10], [11], [12], with $\beta = 1$ (i.e., average for all predictions) or $\beta = 0.05$ (mimicking the "best-of-20" metric used in [10]) being used in this work. As these metrics also does not take into account the severity of different false predictions [25], [26] and requires equally long prediction horizons as well, they are only applied for constant gap sizes ($t_0 = \min\{t \mid t_{\text{C}}(t) - t = \Delta t\} \ll t_{\text{crit}}$). However, due to their similarity, only the *ADE* is discussed further in this work, while the result for *FDE* can instead be found in the supplementary materials.

Similarly to the *FDE*, an additional metric, namely the miss rate (*MR*) is provided in the supplementary materials as well. Furthermore, we propose a novel metric that considers the potential consequences of a wrong prediction.

- *TNR-PR*: The True Negative Rate under Perfect Recall is a metric applied to binary predictions, with the explicit goal to consider the vastly different consequences of false negative and positive predictions at $t_0 \approx t_{\text{crit}}$ (i.e., a potential accident vs unnecessary braking, see Appendix A). To that end, the threshold for classifying a prediction a_{pred} as positive is set as low as necessary to achieve perfect recall on the test set (i.e., there are no false negative predictions). It then estimates the usefulness of prediction models in improving the efficiency of path planning by evaluating their true negative rate (TNR) under this decision threshold. This metric is therefore equivalent to the likelihood that a prediction model can prevent needless braking while guaranteeing safe interactions. As earlier predictions do not necessitate

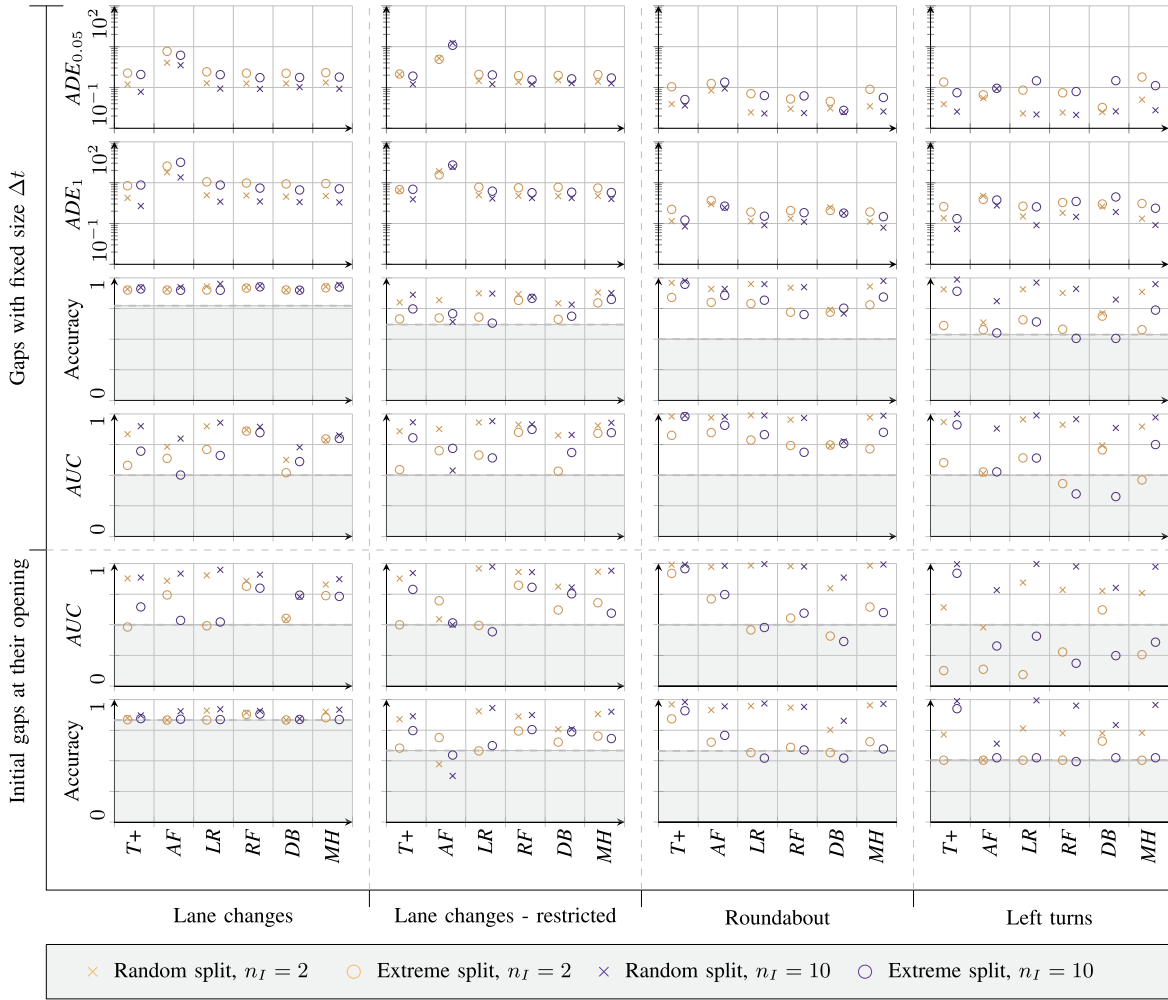


Fig. 4. The results of evaluating behavior prediction models in gap acceptance scenarios on different datasets, with prediction being made either at the initial opening of the gap ($t_0 = t_S$) or with fixed gap sizes ($t_0 = \min\{t \mid t_{\mathcal{C}}(t) - t = \Delta t\}$). The color indicates the number of input time-steps n_I given to the models, and the marker type denotes the splitting method, which can be *random* or *extreme*. The dashed gray lines indicate the performance F_7 of a uniformly random binary predictor. All results can be found in the form of tables in the supplementary materials.

a model with perfect recall, due to available time to wait for further data, this metric is only applied to predictions made at the last possible time (i.e., $t_0 = t_{\text{crit}} - t_\epsilon$).

V. RESULTS

Our benchmark provides insights into performance of tested models under different conditions (Figs. 4 and 5). In this section, we first discuss our findings regarding the prediction performance across models depending on specific prediction problems, which are defined by number of input time steps n_I , the splitting method, the scenarios, and the prediction time t_0 . We will then discuss the differences between individual models.

A. Prediction Problem's Influence on Model Performance

As one may expect, when provided with more input time steps ($n_I = 10$ vs. $n_I = 2$), i.e., more information is provided to extract signs of future behavior from, the models' predictions were generally better. Second, performance of the models tested on the most unintuitive samples (the extreme splitting case) was worse than in the random splitting case. There, the

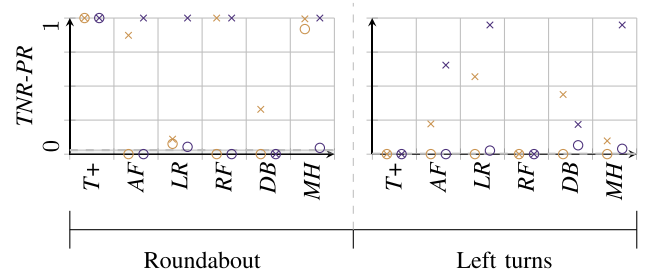


Fig. 5. The true negative rate under perfect recall ($TNR-PR$) of different prediction models, using the same visualizations as in Fig. 4, tested on the last useful predictions ($t_0 = t_{\text{crit}} - t_\epsilon$).

models have to extrapolate to situations outside the training domain, a typically far more difficult task than the interpolation inside the training domain needed for prediction on random test samples [57]. The poor performance on unintuitive samples is especially pronounced when looking at the $TNR-PR$ at critical gaps (Fig. 5), where no model could substantially outperform a random predictor on both datasets.

When comparing the predictions of models on the two lane-change datasets (the two left columns in Fig. 4), it can be expected that the models' performance should be better on the restricted one, due to the removal of many large gaps that were unintuitively rejected. However, this was only observed in the *AUC* metric, while the opposite could be observed for Accuracy (caused by the exploitation of the larger bias towards rejected gaps in the unrestricted lane change dataset, an explanation supported by the high *MR* (supplementary materials) in those cases) and *ADE*. Therefore, it can be assumed that *AUC* is the most reliable metric here, a result which is supported for example by Huang and Ling [58]; this metric will be our main focus from here on.

When comparing the difficulty of different scenarios, it can be seen that, generally, the prediction of human behavior at roundabouts seems to be the easiest, having the best *AUC* values in 30 of 48 cases (each two prediction times, input steps, and splitting methods on six models). A possible explanation here might be the short prediction horizon of less than three seconds (Table II), which leaves less room for the target vehicle to behave unexpectedly, although scenario-specific reasons cannot be excluded. This explanation is also supported by the finding that—at least for $n_I = 10$ —the scenario with the next shortest gap, the left turns, gets the best results out of the remaining three datasets in 12 of 24 cases. This is not the case for $n_I = 2$, where the left-turn dataset is the worst of all four datasets in 15 of 24 cases. This can be explained by the fact that in this scenario, the target vehicle starts in an idling position, from which not much information can be gained. Technically, predictions from both $n_I = 2$ and $n_I = 10$ should be made at the same time, but due to the lack of trajectory data before t_S in this dataset, the predictions are made in fact at $t_0 > t_S + (n_I - 1) * \delta t$. Therefore, prediction for $n_I = 10$ are generally made later, which leaves more room for the onset of motion, making predictions easier.

When comparing the *AUC* at different prediction times t_0 , it can be observed that on those datasets with a lower median $t_C(t_0)$ (Table II) the prediction were better more often than not, in 59 of 96 cases. This again lends credence to the hypothesis that earlier predictions with longer prediction horizons are more challenging. In the case of trajectory prediction models, this seems logical, as a longer prediction horizon leaves more time for prediction errors to propagate and compound on each other. There is also the higher probability that the duration between t_0 and the actual human decision is larger, which leads to a higher probability that there are no indications of that decision in the human behavior yet.

B. Differences Between Models

Comparing the trajectory prediction models, we found that the Trajectron++ model (*T+*) consistently outperforms the AgentFormer model (*AF*). This is surprising, as *AF* previously outperformed *T+* on pedestrian trajectory prediction benchmarks [12]. This contradiction might be explained by over-fitting the many trainable parameters for *AF* on relatively small datasets here. Meanwhile, the logistic regression (*LR*) model is often the most promising approach for binary prediction models (best

AUC value in 18 of 32 cases, and best *TNR-PR* value in 5 out of 8 cases), especially when tested on random samples, where it has the best *AUC* in all possible sixteen cases. Together, those results indicate that increasing the complexity of such models and their number of parameters might not be a panacea, with simpler models being more promising, especially if datasets are relatively small.

When we compare binary models against trajectory prediction models, we can observe differing behavior for different scenarios. On the one hand, the best performance on the lane-change datasets is generally achieved by a binary prediction model (the best *AUC* values come from binary models in all 16 cases), while on the other hand, similar performance can be observed on the other two datasets. One main difference here is the prediction horizon ($\Delta t \approx 10$ s and $\Delta t < 4$ s respectively, see Table II), which might indicate that trajectory prediction models are more impaired by such longer prediction horizon than binary models. That the average displacement errors are much more noticeable in the lane change scenario also supports this explanation, further showing the difficulties of using trajectory prediction under those conditions. However, a deeper analysis into the causes for these observations when comparing models would likely require a number of ablation studies, which is outside the scope of this work.

Lastly, when evaluating the promise of including binary prediction models into trajectory prediction models (relying on the transformation function T_2 from Table I), we can hold that the benefits are mostly negligible if existent at all, except at roundabout and left turns, where some improvements can be seen in average displacement errors. Nonetheless, due to the problems with that metric, more than those results are needed to render a final judgment. However, due to the small size of datasets and the low number of models, the results discussed here should be treated with care.

VI. CONCLUSION

We proposed a framework that connects previously disparate datasets, models, and metrics in the benchmark for testing behavior prediction models in gap acceptance scenarios. We demonstrated its potential and flexibility by comparing two state-of-the-art trajectory prediction models with several binary gap acceptance models. Additionally, we showed that relying on the characteristic time points of gap acceptance scenarios to select the most unintuitive samples in the splitting module is a promising approach to analyzing model generalization, as seen by the general decrease in performance for models trained on those samples. Our framework is open-source and specifically designed in a modular way to simplify adding new datasets, splitting methods, models, and evaluation metrics, which allows researchers to easily expand it in future. This can speed up the testing of new models, only requiring the adaption of the models' implementation to the format of our framework, while previously one would need to write separate code for every model applied to every scenario.

One particularly important addition to the benchmark would be datasets containing more critically accepted gaps, as this

would allow for an increased meaningfulness of metrics applied to last useful predictions. Additionally, a metric better aligned with the main purpose of a prediction model as a part of an autonomous vehicle is still needed. Likewise, currently there exists no method for calculating the similarity between testing and training set ζ ; the future addition of this would permit a quantitative comparison of a model's robustness against unintuitive test samples. Lastly, one could expand the framework to provide scenario-independent inputs similar to $t_{\underline{C}}$, which would enable training a model on two unrelated datasets simultaneously, leading to a better estimation of the models generalizability.

We acknowledge that testing a model in gap acceptance scenarios alone is necessary, but not sufficient for justifying its usage in actual vehicles. Consequently, expanding the framework to non-gap-acceptance scenarios is an important avenue for future research. This will enable more holistic testing, although only for models predicting (and metrics evaluating) trajectories. Nonetheless, we argue that performance of models on non-gap acceptance scenarios should still be given lower priority compared to gap acceptance scenarios which are more safety-critical.

Our results resonate with the recent literature on hybrid AI [59], [60], showing that including binary prediction models in specific scenarios might make data-driven trajectory prediction models more reliable, especially in accurately predicting dangerous situations. However, especially for the unintuitive and safety-critical edge cases, most models often performed only slightly better than a random predictor at best. Therefore, there currently seems to be no model that a trajectory-planning algorithm can rely on to substantially increase the effectiveness of an autonomous vehicle's driving style in every scenario, necessitating further research into such models.

APPENDIX A

APPENDIX DETAILING THE FRAMEWORK FOR BENCHMARKING GAP ACCEPTANCE MODELS

A. Influence of t_0 on the Importance of Predictions

Due to the differing consequences of false negative and false positive predictions when using binary predictions a_{pred} , there are limitations on which metrics are usable at certain t_0 . For $t_0 \ll t_{\text{crit}}$, the consequences of a wrong prediction are generally minor, as time is left to wait for future information before more significant changes to trajectory planning are necessary. Furthermore, even if the target vehicle would immediately accept the gap after t_0 , the necessary response is likely neither uncomfortable nor risky. Consequently, symmetric metrics can be used here.

For $t_0 \approx t_{\text{crit}}$ however, no more time for further observations is left, resulting in far more severe consequences for both false positive and false negative predictions. A false positive prediction would unnecessarily result in a harsh and uncomfortable braking maneuver. Meanwhile, a wrong negative prediction leads to an unsafe gap acceptance maneuver, with the safety of the interaction between V_E and V_T no longer in the control of the autonomous vehicle V_E . Accidents or the need for dangerous emergency maneuvers, which could result in material damage or even bodily harm, are then possible. As the latter should be

avoided at all costs, a false negative prediction at this time is far more consequential, which should be reflected in the evaluation metric.

B. Influence of t_0 on the Size of the Dataset

The method for determining the time t_0 can impact the size of the resulting dataset (Table II) due to the condition from Equation (3). Namely, for constant gap size ($t_0 = \min\{t | t_{\underline{C}}(t) - t = \Delta t\}$), the number of available samples will be reduced, as all gaps with an initial smaller gap size ($t_{\underline{C}}(t_S) - t_S < \Delta t$) will be excluded. The same is the case for gaps already accepted before t_0 . For critical gap sizes instead (i.e., $t_0 = t_{\text{crit}} - t_{\epsilon}$), all gaps accepted before t_{crit} will be excluded, leading to extremely biased datasets, sometimes even removing all accepted gaps.

C. Detailing the Timing Prediction

The predicted time $t_{A,\text{pred}}$ is expressed using the decile values:

$$t_{A,\text{pred}} = Q_9(\mathcal{T}_A) = \{Q_{\mathcal{T}_A}(p) | p \in \{0.1, 0.2, \dots, 0.9\}\} \in \mathbb{R}^9 \quad (\text{A.6})$$

Here, $Q_{\mathcal{T}_A}$ is the quantile function associated with this underlying distribution of \mathcal{T}_A , which might for example be expressed as a set of individual time points t_A . This means that with a likelihood of p , $t_A < Q_{\mathcal{T}_A}(p)$ will be the case, if the human decides to accept the gap ($t_A < t_C$):

$$P(t_A < Q_{\mathcal{T}_A}(p) | t_A < t_C) = p$$

As the likelihood of accepting the gap is given by a_{pred} , one can get:

$$\begin{aligned} P(t_A < Q_{\mathcal{T}_A}(p)) &= P(t_A < Q_{\mathcal{T}_A}(p) | t_A < t_C) P(t_A < t_C) \\ &= p a_{\text{pred}} \end{aligned}$$

As t_A is predicted under the assumption that the gap is accepted, every decile value in $t_{A,\text{pred}}$ should be smaller than t_C and larger than t_S .

D. Transforming the Predictions

When implementing the transformation of a prediction d_{pred} into another form, two instances of the trajectory prediction model *Trajectron++* [10] are used, namely M_A , trained on all samples from the specific dataset D (D_I and D_O) where $a = 1$, and M_{-A} , trained on the remaining samples where $a = 0$. Furthermore, the function f_a is defined, which extracts the gap acceptance decision from a single predicted trajectory, using T_A from (2) (assuming $t_C \approx \max T_O$ based on Equations (4) and (5)).

$$f_a(\mathbf{X}_{T_O,p}) = \begin{cases} 1 & T_A(\mathbf{X}_{T_O,p}) < \max T_O \\ 0 & \text{else} \end{cases} \quad (\text{A.7})$$

T_1 : The use of f_a and Equation (A.6) results in

$$\begin{aligned} a_{\text{pred}} &= \frac{1}{n_p} \sum_p f_a(\mathbf{X}_{T_O,p}) \\ t_{A,\text{pred}} &= Q_9(\{T_A(\mathbf{X}_{T_O,p}) | f_a(\mathbf{X}_{T_O,p}) = 1\}) \end{aligned} \quad (\text{A.8})$$

T_2 : M_A and M_{-A} are used to respectively predict two sets of trajectories, namely $\mathbf{X}_{T_{O,pred,A}}$ and $\mathbf{X}_{T_{O,pred,-A}}$. The final set of trajectories $\mathbf{X}_{T_{O,pred}}$ is assembled from select trajectories out of these two sets:

$$\mathbf{X}_{T_{O,pred}} = \{\mathbf{X}_{T_{O,p,-A}} | p \in \mathbf{R}_{-A}\} \cap \{\mathbf{X}_{T_{O,p,A}} | p \in \mathbf{R}_A\}.$$

This selection is based on the random selection function $R(m, M, \mathbf{W})$, which randomly selects m samples from a set M , with the probabilities of selection being proportional to the weights \mathbf{W} . Only samples that actually represent the desired decision are viable (e.g., in samples from $\mathbf{X}_{T_{O,p,A}}$, the gap must be accepted):

$$\mathbf{R}_{-A} = R(n_p(1 - a_{pred}), \{p | f_a(\mathbf{X}_{T_{O,p,-A}}) = 0\}, \mathbf{1})$$

$$\mathbf{R}_A = R(n_p a_{pred}, \{p | f_a(\mathbf{X}_{T_{O,p,A}}) = 1\}, \mathbf{W}_A).$$

Here, \mathbf{W}_A is chosen so that the distribution of t_A described by the decile values $t_{A,pred,i}$ of $t_{A,pred}$ (Equation (A.6)) is maintained:

$$\sum_{\{p | t_{A,pred,i} < T_A(\mathbf{X}_{T_{O,p}}) < t_{A,pred,i+1}\}} w_{A,p} = 1 \quad \forall i$$

This approach of using conditional trajectory prediction models is inspired by Xie et al. [42] and Hu et al. [40].

T_3 : Here, one uses M_A to get generate $\mathbf{X}_{T_{O,pred,A}}$, based on which one can get

$$t_{A,pred} = Q_9(\{T_A(\mathbf{X}_{T_{O,p,A}}) | f_a(\mathbf{X}_{T_{O,p,A}}) = 1\}).$$

REFERENCES

- [1] D. Holland-Letz, M. Kässer, B. Kloss, and T. Müller, "Mobility's future: An investment reality check mckinsey," May 2022. [Online]. Available: <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mobilitys-future-an-investment-reality-check>
- [2] J. S. Brar and B. Caulfield, "Impact of autonomous vehicles on pedestrians' safety," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 714–719.
- [3] J. Meyer, H. Becker, P. M. Bösch, and K. W. Axhausen, "Autonomous vehicles: The next jump in accessibilities?," *Res. Transp. Econ.*, vol. 62, pp. 80–91, 2017.
- [4] J. Pizarov and G. Mester, "The future of autonomous vehicles," *FME Trans.*, vol. 49, no. 1, pp. 29–35, 2021.
- [5] M. Milford, S. Anthony, and W. Scheirer, "Self-driving vehicles: Key technical challenges and progress off the road," *IEEE Potentials*, vol. 39, no. 1, pp. 37–45, Jan./Feb. 2020.
- [6] J. Wang, L. Zhang, Y. Huang, and J. Zhao, "Safety of autonomous vehicles," *J. Adv. Transp.*, vol. 2020, 2020, Art. no. e8867757.
- [7] A. Sinha, S. Chand, V. Vu, H. Chen, and V. Dixit, "Crash and disengagement data of autonomous vehicles on public roads in california," *Sci. Data*, vol. 8, 2021, Art. no. 298.
- [8] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions," *Robot.: Sci. Syst.*, vol. 2, pp. 1–9, 2016.
- [9] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 33–47, Jan. 2022.
- [10] T. B. Salzmann, P. I. Chakravarty, and M. Pavone, "Trajectron : Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 683–700.
- [11] F. Giuliani, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2021, pp. 10335–10342.
- [12] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9813–9823.
- [13] F. Diehl, T. Brunner, M. T. Le, and A. Knoll, "Graph neural networks for modelling traffic participant interaction," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 695–701.
- [14] S. Kolekar, J. de Winter, and D. Abbink, "Human-like driving behaviour emerges from a risk-based driver model," *Nat. Commun.*, vol. 11, 2020, Art. no. 4850.
- [15] X. Huang et al., "HYPER: Learned hybrid trajectory prediction via factored inference and adaptive sampling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 2906–2912.
- [16] A. Cui, S. Casas, A. Sadat, R. Liao, and R. Urtasun, "LookOut: Diverse multi-future prediction and planning for self-driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16107–16116.
- [17] R. Chandra, A. Bera, and D. Manocha, "Using graph-theoretic machine learning to predict human driver behavior," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2572–2585, Mar. 2022.
- [18] L. Sun, X. Jia, and A. Dragan, "On complementing end-to-end human behavior predictors with planning," *Robot. Sci. Syst.*, Jan. 2021.
- [19] Z. Cao, E. Biyik, G. Rosman, and D. Sadigh, "Leveraging smooth attention prior for multi-agent trajectory prediction," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2022, pp. 10723–10730.
- [20] A. Bighashdel, P. Meletis, and G. Dubbelman, "Towards equilibrium-based interaction modeling for pedestrian path prediction," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–8.
- [21] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezaatofghi, and S. Savarese, "SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1349–1358.
- [22] P. Kothari, B. Siffringer, and A. Alahi, "Interpretable social anchors for human trajectory forecasting in crowds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15551–15561.
- [23] F. Camara et al., "Pedestrian models for autonomous driving part II: High-level models of human behavior," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5453–5472, Sep. 2021.
- [24] J. Yue, D. Manocha, and H. Wang, "Human trajectory prediction via neural social physics," in *Proc. Eur. Conf. Comput. Vis.*, Jul. 2022, pp. 376–394.
- [25] B. Ivanovic and M. Pavone, "Injecting planning-awareness into prediction and detection evaluation," in *Proc. IEEE Intell. Veh. Symp.*, 2022, pp. 821–828.
- [26] A. Farid, S. Veer, B. Ivanovic, K. Leung, and M. Pavone, "Task-relevant failure detection for trajectory predictors in autonomous vehicles," 2022, *arXiv:2207.12380*.
- [27] A.R. Srinivasan et al., "Comparing merging behaviors observed in naturalistic data with behaviors generated by a machine learned model," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2021, pp. 3787–3792.
- [28] A.R. Srinivasan et al., "Beyond RMSE: Do machine-learned models of road user interaction produce human-like behavior?," 2022, *arXiv:2206.11110*.
- [29] S. Noh, "Probabilistic collision threat assessment for autonomous driving at road intersections inclusive of vehicles in violation of traffic rules," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4499–4506.
- [30] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On Adversarial robustness of trajectory prediction for autonomous vehicles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15159–15168.
- [31] G. Markkula et al., "Defining interactions: A conceptual framework for understanding interactive behaviour in human and automated road traffic," *Theor. Issues Ergon. Sci.*, vol. 21, no. 6, pp. 728–752, 2020.
- [32] S. K. Jayaraman, L. P. Robert, X. J. Yang, and D. M. Tilbury, "Multimodal hybrid pedestrian: A hybrid automaton model of urban pedestrian behavior for automated driving applications," *IEEE Access*, vol. 9, pp. 27708–27722, 2021.
- [33] J. Pekkanen et al., "Variable-drift diffusion models of pedestrian road-crossing decisions," *Comput. Brain Behav.*, vol. 5, pp. 60–80, 2022.
- [34] A. Theofilatos, A. Ziakopoulos, O. Oviedo-Trespalacios, and A. Timmis, "To cross or not to cross? Review and meta-analysis of pedestrian gap acceptance decisions at midblock street crossings," *J. Transport Health*, vol. 22, 2021, Art. no. 101108.
- [35] D. Yang, H. Zhang, E. Yurtsever, K. A. Redmill, and U. Özgüner, "Predicting pedestrian crossing intention with feature fusion and spatio-temporal attention," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 221–230, Jun. 2022.
- [36] S. Mafi, Y. Abdelrazig, and R. Doczy, "Analysis of gap acceptance behavior for unprotected right and left turning maneuvers at signalized intersections using data mining methods: A driving simulation approach," *Transp. Res. Rec.*, vol. 2672, pp. 160–170, 2018.

- [37] A. Abhishek, M. A. A. Boon, and M. Mandjes, "Generalized gap acceptance models for unsignalized intersections," *Math. Methods Oper. Res.*, vol. 89, pp. 385–409, 2019.
- [38] A. Zgonnikov, D. Abbink, and G. Markkula, "Should I stay or should I go? cognitive modeling of left-turn gap acceptance decisions in human drivers," *Hum. Factors*, 2022, Art. no. 00187208221144561.
- [39] M. Arafat, M. Hadi, T. Hunsanon, and K. Amine, "Stop sign gap assist application in a connected vehicle simulation environment," *Transp. Res. Rec.*, vol. 2675, pp. 1127–1135, 2021.
- [40] Y. Hu, X. Jia, M. Tomizuka, and W. Zhan, "Causal-based time series domain generalization for vehicle intention prediction," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 7806–7813.
- [41] E. Balal and R. L. Cheu, "Comparative evaluation of fuzzy inference system, support vector machine and multilayer feed-forward neural network in making discretionary lane changing decisions," *Neural Netw. World*, vol. 28, no. 4, pp. 361–378, 2018.
- [42] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He, "A data-driven lane-changing model based on deep learning," *Transp. Res. Part C: Emerg. Technol.*, vol. 106, pp. 41–60, 2019.
- [43] A. Das, M. N. Khan, and M. M. Ahmed, "Nonparametric multivariate adaptive regression splines models for investigating lane-changing gap acceptance behavior utilizing strategic highway research program 2 naturalistic driving data," *Transp. Res. Rec.*, vol. 2674, pp. 223–238, 2020.
- [44] B. Khelifa, I. Ba, and A. Tordeux, "Predicting highway lane-changing maneuvers: A benchmark analysis of machine and ensemble learning algorithms," *Physica A: Stat. Mech. Appl.*, vol. 612, Feb. 2023, Art. no. 128471.
- [45] S. Mozaffari, E. Arnold, M. Dianati, and S. Fallah, "Early lane change prediction for automated driving systems using multi-task attention-based convolutional neural networks," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 758–770, Sep. 2022.
- [46] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Robot. Res.*, vol. 39, pp. 895–935, 2020.
- [47] B. R. Kadali, P. Vedagiri, and N. Rathi, "Models for Pedestrian Gap Acceptance Behaviour Analysis At Unprotected Mid-Block Crosswalks Under Mixed Traffic Conditions," *Transp. Res. Part F: Traffic Psychol. Behav.*, vol. 32, pp. 114–126, 2015.
- [48] R. Yao, W. Zeng, Y. Chen, and Z. He, "A deep learning framework for modelling left-turning vehicle behaviour considering diagonal-crossing motorcycle conflicts at mixed-flow intersections," *Transp. Res. Part C: Emerg. Technol.*, vol. 132, 2021, Art. no. 103415.
- [49] 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, 2017.
- [50] M. Yang, X. Wang, and M. Qudus, "Examining lane change gap acceptance, duration and impact using naturalistic driving data," *Transp. Res. Part C: Emerg. Technol.*, vol. 104, pp. 317–331, 2019.
- [51] E. A. I. Pool, J. F. P. Kooij, and D. M. Gavrila, "Crafted vs learned representations in predictive models—a case study on cyclist path prediction," *IEEE Trans. Intell. Veh.*, vol. 6, no. 4, pp. 747–759, Dec. 2021.
- [52] H. Müller, W. Müller, S. Marchand-Maillet, T. Pun, and D. M. Squire, "A framework for benchmarking in CBIR," *Multimed. Tools Appl.*, vol. 21, pp. 55–73, 2003.
- [53] M. Zaffar et al., "VPR-Bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change," *Int. J. Comput. Vis.*, vol. 129, pp. 2136–2174, 2021.
- [54] Y. Cao, C. Xiao, A. Anandkumar, D. Xu, and M. Pavone, "AdvDO: Realistic adversarial attacks for trajectory prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 36–52.
- [55] R. Krajewski, J. Bock, L. Kloecker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2118–2125.
- [56] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The round dataset: A drone dataset of road user trajectories at roundabouts in germany," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–6.
- [57] E. Barnard and L. Wessels, "Extrapolation and interpolation in neural network classifiers," *IEEE Control Syst. Mag.*, vol. 12, no. 5, pp. 50–53, Oct. 1992.
- [58] J. Huang and C. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 299–310, Mar. 2005.
- [59] G. Marcus, "The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence," 2020, *arXiv: 2002.06177*.
- [60] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, and A. t. Teije, "Modular design patterns for hybrid learning and reasoning systems," *Appl. Intell.*, vol. 51, pp. 6528–6546, Sep. 2021.



Julian F. Schumann received the master's degree in 2021 in mechanical engineering from the TU Delft, Delft, The Netherlands, where he has been working toward the Ph.D. degree focusing on hybrid-AI models for behavior prediction of human traffic participants for use in automated vehicles since 2021. His research interests include modeling of human behavior, meaningful evaluation of such models, and the integration of knowledge-based and machine-learned models.



Jens Kober received the Ph.D. degree in engineering from TU Darmstadt, Darmstadt, Germany, in 2012. He is currently an Associate Professor with TU Delft, Delft, The Netherlands. He was a Postdoctoral Scholar jointly with CoR-Lab, Bielefeld University, Bielefeld, Germany, and with Honda Research Institute Europe, Offenbach, Germany. He was the recipient of the 2018 IEEE RAS Early Academic Career Award and the 2022 RSS Early Career Award. His research interests include motor skill learning, imitation learning, interactive learning, and machine learning for control.



Arkady Zgonnikov received the Ph.D. degree from the University of Aizu, Aizuwakamatsu, Japan, in 2014. Since 2014, he has been a Postdoctoral Researcher with the University of Galway, Galway, Ireland (funded by Irish Research Council) and Delft University of Technology, Delft, The Netherlands (funded by the AiTech initiative). Since 2020, he has been an Assistant Professor with the Delft University of Technology. His research interests include responsible AI and cognitive modeling, and the applications of those to human-robot interactions in traffic and beyond.