Cyclist Motion State Forecasting – Going beyond Detection

Maarten Bieshaar, Stefan Zernetsch, Katharina Riepe, Konrad Doll, and Bernhard Sick

***Abstract*— In this article, we present two novel methods to forecast the motion states of cyclists. The states we aim to anticipate are waiting, starting, moving, and stopping. This information can be utilized to increase road safety when used in an automated vehicle. We classify the cyclist motion state for every step in a discrete-time horizon using a single neural network in the first method. In our second approach, we consider a two-stage model, i.e., a neural network predicts the current and the next motion state, and a second quantile regression neural network (QRNN) forecasts the time to transition between these two motion states. Our results show that both methods have advantages and disadvantages. The first method can forecast multiple changes in motion state while the second is restricted to a single transition. However, the first method is limited to a fixed forecast horizon. The two-stage approach, which forecasts motion and time separately, is more flexible regarding the forecast horizon, i.e., it can forecast very long as well as short time spans. Regarding the transition detection performance, both methods perform equally well. Our experiments show that the time to transition to the next motion state can be forecasted accurately, especially for short-time horizons.**

I. Introduction

In recent decades, great efforts have been made to reduce the number of road users injured or killed in road accidents. While the number of wounded drivers of motorized vehicles has decreased significantly, the number of injuries to vulnerable road users (VRU) such as pedestrians and cyclists has stagnated [7]. In Germany, the number of fatally injured cyclists has even risen recently [23]. This is partly due to the rising number of cyclists in road traffic and partly because cyclists benefit less from passive safety features in vehicles than occupants of motorized vehicles. To counteract this trend, many researchers and car manufacturers are currently intensifying their work on driver assistance systems that actively protect VRU, e.g., by warning the driver of a vehicle or the vehicle automatically performing braking maneuvers. For these driver assistance systems, it is necessary to know the position of the VRU and forecast their future behavior over a period of 2-3 seconds [10].

In this article, we present two methods to forecast the motion state of cyclists. In addition to detecting the current motion state, the goal is to forecast the transition to the following motion state. Thus, potentially dangerous situations, such as the intention of a cyclist waiting at the side of the road to enter the road (e.g., Fig. 1), can be recognized early, and an appropriate measure can be taken in time.

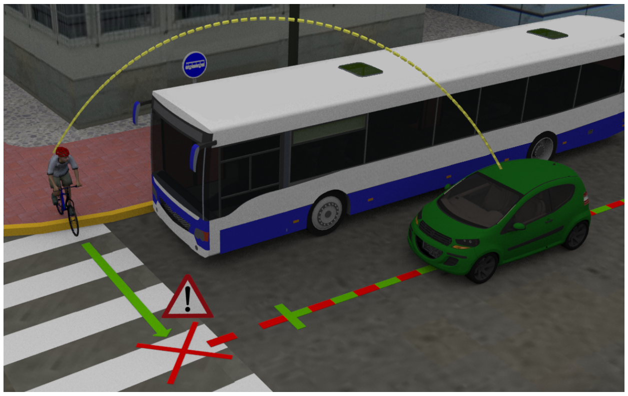


Fig. 1: Cyclist starting to cross the road in front of bus.

A. Main Contributions and Outline of this Paper

Our main contributions are two methods to **forecast the future motion states** of cyclists. While existing methods focus on either detection of the current motion state or forecasting the future trajectory of cyclists, our main goal is to anticipate, i.e., forecast, the future motion state, which to our knowledge, has not been investigated at the time of writing.

We present two different approaches. The first approach classifies motion states for every future time step within a forecast horizon using a single neural network. The second approach uses a neural network to predict the transition between the current and the following motion state, and a second quantile regression neural network (QRNN) to estimate the time until the transition occurs.

By evaluating our methods using a large dataset recorded at an urban intersection, we demonstrate both methods' advantages and disadvantages. Moreover, we show that both methods can be used in real-world traffic applications.

The remainder of this article is structured as follows: In Section II, we present related work. Section III describes our methodology to forecast the cyclist motion state. The data acquisition and evaluation methodology are presented in Section IV, and the experimental results are reviewed in Section V. Lastly, in Section VI, the main conclusions and open issues for future work are reviewed.

II. Related Work

In this section, we present different approaches to detect and anticipate the behavior of VRUs. Over the last years, VRU intention detection has become an active field of

M. Bieshaar, K. Riepe, and B. Sick are with the Intelligent Embedded Systems Lab, University of Kassel, Kassel, Germany {mbieshaar | bsick}@uni-kassel.de, uk037924@student.uni-kassel.de

S. Zernetsch and K. Doll are with the Faculty of Engineering, University of Applied Sciences Aschaffenburg, Aschaffenburg, Germany {stefan.zernetsch | konrad.doll}@th-ab.de

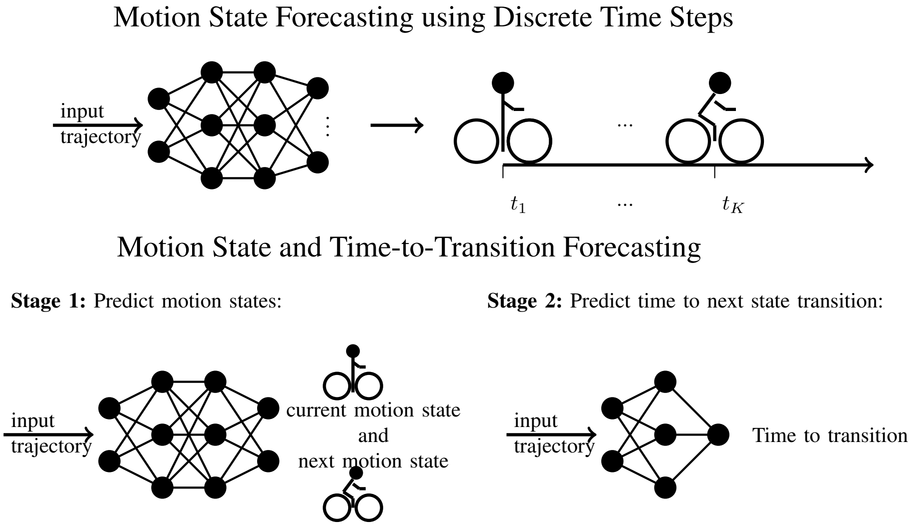

Motion State Forecasting using Discrete Time Steps
input
trajectory
t_1
...
t_K
Motion State and Time-to-Transition Forecasting
Stage 1: Predict motion states:
Stage 2: Predict time to next state transition:
input
trajectory
current motion state
and
next motion state
input
trajectory
Time to transition

Fig. 2: Schematics of the two different approaches for forecasting cyclist motion states.

research. Intention detection can be roughly categorized into the detection of VRU motion states and the forecasting of VRU trajectories.

Motion State Detection: The goal of motion state detection is to estimate the VRU's current state of motion, such as waiting or moving. It is usually performed in every time step. In [12], the authors compare four methods to distinguish whether a pedestrian stops at the curbside or continues walking. Two methods based on Kalman filters using information about the pedestrians' past positions are compared to two methods based on dynamic Gaussian Process Models (GPDM) and hierarchical trajectory matching using features derived from dense optical flow. A convolutional neural network is used in [20] to detect whether pedestrians are standing or walking and to detect their orientation using information from image data. In [14], motion history image-based histogram feature vectors extracted from stereo-image sequences in combination with a support-vector machine are used to detect the motion states, i.e., starting, stopping, waiting, and moving, of pedestrians. In [10], the pedestrians' past head velocities, in combination with a multilayer perceptron (MLP), are used to detect starting or waiting. While there are many attempts regarding motion state detection of pedestrians, there are only a few articles about cyclists' motion detection. In [25], the authors adapt the approach from [14] to detect whether a cyclist is waiting or starting. The authors from [16] make use of 3D poses of cyclists to distinguish between the motion states waiting and starting. Further, motion state detection approaches can be found in [2], [8], [18].

The above mentioned methods have in common that the goal is to detect motion states or transitions between motion states at the current time. In this work, we focus on forecasting the time until a transition between states occurs.

Trajectory Forecast: The goal of VRU trajectory forecasts is to estimate the VRU's future positions. It has become an active field of research over the past years. While the methods create a different output compared to motion state detection, similar methods and input features can be used. In addition to motion state transition, the authors from [12] use their approaches to forecast pedestrian trajectories for sub-second time intervals. In [10], a slightly altered version of the MLP applied for motion state detection is used to forecast pedestrian and cyclist trajectories 2.5 seconds into the future. In [19], the authors introduce a trajectory forecast based on pedestrians' body joint trajectories in combination with balanced GPDM to forecast trajectories up to one second into the future. Other approaches are presented by Pool et al. in [6] and Kooji et al. [15], in which the authors propose methods for cyclist path prediction using the local road topology and other additional context information.

While these methods all create forecasts about the future movements of VRU, only positions are forecasted, and no statement is made about the VRU's future motion state. Therefore, we see this article as an approach to close the gap between motion state detection and trajectory forecast.

III. Methodology

In this article, we investigate two different approaches for predicting the future motion state of a cyclist. A schematic of both approaches is depicted in Fig. 2. Both approaches share a common characteristic because they both use features based on the cyclist's past trajectory as input [24]. However, the approaches could also be adapted to use different input features like image sequences of human pose trajectories. The first method involves a multi-task learning approach [21]. For this purpose, we adopt a time-discrete modeling of the forecasting lead times $t_1, \ldots, t_K$, where t_K is the maximum forecasting horizon. We regard the motion state's prediction at each lead time t_k as a classification problem. Therefore, we use a neural network with one output head for each lead time, i.e., multi-task learning. The first approach is referred to as *Motion State Forecasting using Discrete-Time Steps*.

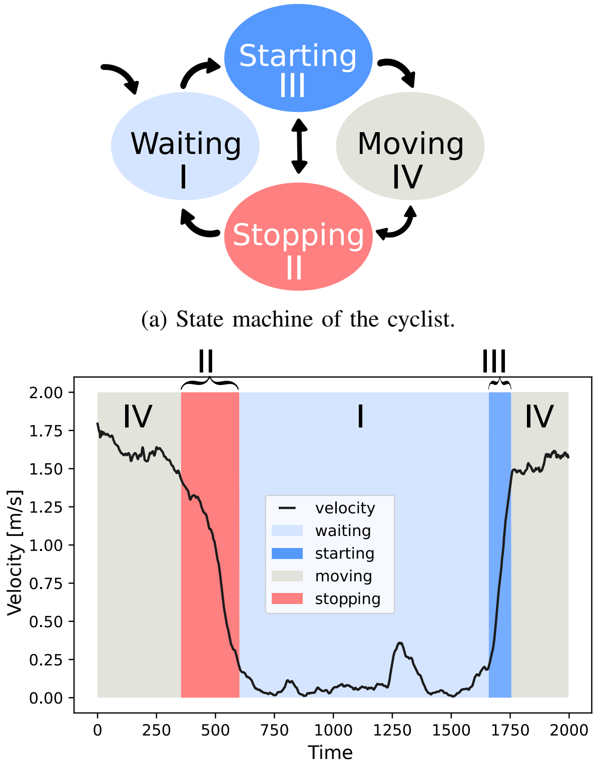
One of the major limitations of the approach is the temporal discretization of the forecast and the fixed forecast horizon, set at design time. Therefore, in the second approach, we pursue a different modeling. It can be described as a two-stage procedure. In the first stage, we again consider a multi-task multi-class problem. But, here, we have only two output heads. One for the current motion state and one for the next motion state. In the second stage, we try to predict the time until the transition from the current state to the next state with a second neural network, i.e., we try to predict the time to transition. The latter can be viewed as a regression problem. The second approach has the advantage that we can have arbitrary short and long prediction time horizons. For modeling the uncertainty regarding the time to transition, we use quantile regression methods, i.e., quantile regression neural networks (QRNN) [4]. We refer to the second approach as *Motion State and Time-to-Transition Forecasting*.

A. Cyclist Behavior Model

In this article, we describe the cyclist motion using a state machine (cf. Fig. 3a) consisting of four states: *Starting*, *Moving*, *Stopping*, and *Waiting*. In the following, we detail the discrete modeling of the motions states. In the *Waiting* state, the cyclist stands still and remains in a fixed position. Minor body movements, e.g., turning of the head, are allowed. We define the state by the movement of the bicycle's rear wheel, i.e., the *Waiting* state begins with the rear wheel's last movement and ends with the first movement of the rear wheel. This, in turn, is the beginning of the *Starting* state. It can be best described as the phase of acceleration from an initial movement of the rear wheel until the cyclist no longer accelerates and moves at an approximately constant velocity. We identify the end of the *Starting* state by a threshold on the cyclist's acceleration, i.e., here of $0.2\,\mathrm{m/s^2}$. The *Moving* state is the phase of nearly constant velocity between the *Starting* and *Stopping* state. Note, the *Moving* state might also comprise phases of acceleration and deceleration. We define the *Stopping* state as the phase of deceleration from the *Moving* state to complete standstill, i.e., the rear wheel's last visible movement. An exemplary motion state sequence with a stopping, subsequently waiting, and starting cyclist including velocity is depicted in Fig. 3b.

B. Motion State Forecasting using Discrete Time Steps

In the following, we detail the motion state forecasting using discrete time steps. With this method, we use a neural network that predicts the motion state at a fixed set of discrete lead times $t_1, \ldots, t_K$. In total, we have K discrete lead times. We use a deep feed-forward, fully-connected neural network to predict the motion states. We use one output layer for each lead time. The output layers are also referred to as heads of the neural network. Every head contains four output neurons for each of the four motion states. We obtain valid probabilities by applying a softmax activation function to each output head. For training, we use the cross-entropy loss function. Training the network can be seen as a multi-task, i.e., multi-label, problem because we have to predict the motion states for each of the K discrete lead times. The multi-label cross-entropy loss of the i-th sample is given by



(a) State machine of the cyclist.

(b) Labeling of motion states.

Fig. 3: Labeling of the cyclist states based on the trajectory. Waiting: The cyclist is standing in a place (moving of the upper body or head is possible). Starting: The cyclists starts to move from a waiting phase until a state is reached in which no or only a little acceleration takes place. Moving: The cyclist is moving, continually. Stopping: The cyclist starts to decelerate and finally stops.

$$L(y_i, \hat{p}_i) = -\sum_{k=1}^{K}\sum_{c=1}^{4} y_{i,t_k}^{(c)} \log(\hat{p}_{i,t_k}^{(c)}), \tag{1}$$

where $y_i \in \mathbb{R}^{K\times 4}$ denotes a tensor whose rows are the one-hot encoded ground truth motion state of the i-th sample at all lead times. Moreover, $\hat{p}_i \in \mathbb{R}^{K\times 4}$ refers to a tensors whose rows are the one-hot encoded predicted probabilities of the neural network for the i-th sample at all lead times. $y_{i,t_k}^{(c)} \in \mathbb{R}^4$ and $\hat{p}_{i,t_k}^{(c)} \in \mathbb{R}^4$ denote the entries belonging to the c-th class (i.e., motion state) at lead time t_k of the ground truth and predicted probability, respectively. We determine the neural network's hyperparameter, e.g., network architecture, activation function, learning rate, experimentally using a grid search. We estimate the time to transition to the next motion state by evaluating the predicted motion state (i.e., maximal class probabilities) of the neural network. Hence, the time to transition equals the time to a switch in the predicted classes. This corresponds to a deterministic point estimate of the time to transition.

C. Motion State and Time-to-Transition Forecasting

In this method, we model the motion state forecasting task using two stages. In the first stage, we predict the current as well as the next motion state. We make no assumptions or statements about when the cyclist will transition to the next motion state in this stage. In the second stage, we aim to forecast the time to transition to the next state. Classification of the current and next state, and the prediction of the time to transition are done with two separate neural networks. The first network which is responsible for predicting the current and next motion state is also referred to as *motion state prediction network*. The second network is referred to as *time-to-transition network*.

The motion state prediction network is a feed-forward, fully-connected neural network with two output heads having four output neurons (one neuron for each motion state). Again, we have a multi-task problem. The network's training is widely done in analogy to the procedure described in Section III-B. Instead of K lead times, we only have two "lead times", i.e., outputs.

The time-to-transition network is also a feed-forward network. The neural network's goal is to predict a continuous quantity, namely the time to the next transition. It is thus a regression task. We use probabilistic forecasting to account for the uncertainty [9]. In our case, probabilistic forecasting is about issuing predictive distributions over the cyclist's time to transition. We use quantile regression (QR), as introduced by Koenker and Basset [13]. QR is widely used in many domains, mainly because of its ability to represent arbitrary predictive distributions in a non-parametric way. The predictive distribution is represented by a discrete set of quantile levels, e.g., 5 % 10 %, 20 %, . . . 80 %, 90 %, 95 %, and 99 % quantile. The integration of QR with neural networks is referred to as QRNN [4]. The QRNN has a dedicated output neuron for each quantile level to represent the predictive distribution.

To train the time-to-transition network, i.e., the QRNN, we use the quantile loss (also referred to as pinball loss or absolute tiled value function). The quantile loss is an asymmetric weighting of positive and negative prediction errors using a tilted form of the absolute value error function. It is defined for a particular quantile value τ. Then, given an observed time to transition t_{o_i} and the predicted value of the τ-th quantile $\hat{y}_i$ of the i-th sample, the quantile loss is defined as follows

$$s_\tau\left(t_{o_i},\hat{y}_i\right)=\begin{cases}\tau|\hat{y}_i-t_{o_i}|, & (\hat{y}_i-t_{o_i})\geq 0\\ (1-\tau)\left|\hat{y}_i-t_{o_i}\right|, & \text{else.}\end{cases} \tag{2}$$

We use this loss function to adjust the neural network's parameters, i.e., one loss function for each discrete quantile level. The issued predictions of the neural network correspond to the τ-th quantiles of the underlying predicted time-to-transition distribution. In contrast to the first method (cf. Section III-B), this time-to-transition estimate is probabilistic.

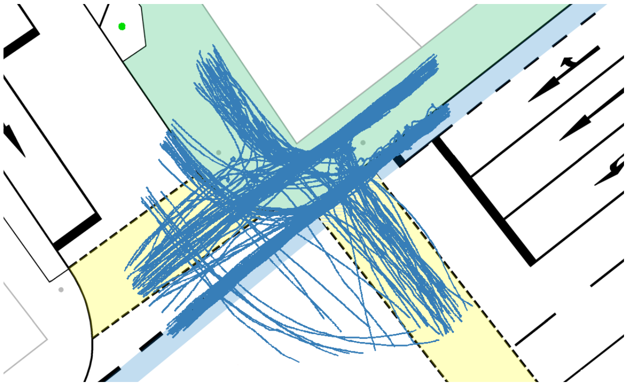


Fig. 4: Trajectories recorded at the research intersection.

IV. Data Acquisition and Evaluation

A. Data Acquisition

To train and evaluate our methods, we use the data set from [1], which was created at a public research intersection at the University of Applied Sciences Aschaffenburg in Germany. The data set consists of cyclist head trajectories recorded using a wide-angle stereo camera system with a frame rate of 50 fps. The cyclist head positions are used since studies have shown that they can serve as an early indicator to detect cyclist intentions [11]. The data set consists of 1329 trajectories of cyclists crossing the intersection (see Fig. 4). For each trajectory, the motion state labels *Waiting*, *Starting*, *Stopping*, and *Moving* are provided. The beginning and end of the state *Waiting* are annotated manually, where the beginning is marked after the rear wheel of the stopping bicycle comes to a halt, and the end is marked when the rear wheel starts moving again when the cyclist starts. The remaining states are labeled automatically (cf. Fig. 3). The states *Starting* and *Stopping* are labeled during the acceleration or deceleration phases after or before the *Waiting* state. *Moving* is labeled between *Starting* and *Stopping*, while the cyclist moves with nearly constant velocity. The trajectories are split into samples consisting of input positions and output classes for model training and evaluation. The output samples for the *Motion Forecasting using Discrete Time Steps* consists of a tensor of size 4×125, i.e., the one-hot encoded active motion state over the next $2.5\,\mathrm{s}$ sampled with $50\,\mathrm{Hz}$. The output samples for the *Motion State and Time-to-Transition Forecasting* consists of the currently active motion state, the state to which the cyclist will transition, and the time between the two states. The distribution of motion states, i.e., classes, for training the neural networks is depicted in Fig. 5. *Waiting* is over-represented due to long waiting times at the traffic lights.

B. Evaluation

For the evaluation, we differentiate between the two methods. For *Motion Forecasting using Discrete Time Steps* approach, we evaluate the 125 forecasted time steps using F_1 and Brier score as metrics for classification performance (cf. [2] for details). We use the same metrics for the *Motion State and Time-to-Transition* approach; however, instead of

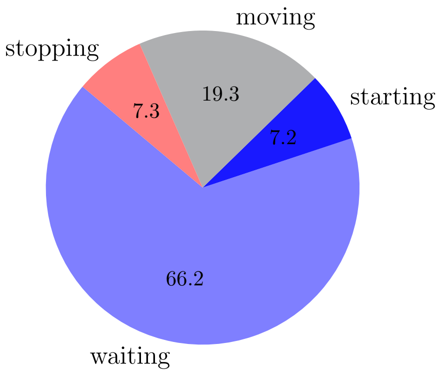


Fig. 5: Class distribution of motion states in data set.

discrete time steps, we evaluate the detection of the current and the forecast of the next state. Additionally, we calculate the error of the forecasted and actual time to transition between states.

The F_1 score is used to evaluate how well the method can distinguish between different motion states, taking into account true positive (tp), false positive (fp), and false negative (fn) classifications, and is given by

$$\mathrm{F}_1 = \frac{\mathrm{tp}}{\mathrm{tp} + 0.5 \cdot (\mathrm{fp} + \mathrm{fn})}. \tag{3}$$

In addition to the F_1 score, the Brier score measures the quality of the probability of the classified value. The Brier score is mainly used for binary problems and deals with the predicted probabilities. The Brier score is negatively oriented, meaning that a smaller Brier score is better than a larger [3]. We calculate a separate Brier score for each state c:

$$\mathrm{BS}_c = \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{k=1}^{K} (\hat{p}_{i,t_k}^{(c)} - y_{i,t_k}^{(c)})^2, \tag{4}$$

with K being the index of the maximal lead time t_K, and $\hat{p}_{i,t_k}^{(c)}$ and $y_{i,t_k}^{(c)}$ as the probability and ground truth value of the i-th sample, the c-th state with lead time t_k. Let, N denote the total number of samples.

Reliability refers to whether the issued predictive distribution is correct, i.e., whether the estimated frequency of the predictive distribution matches the observed frequency.

A probabilistic forecast with probability P is reliable if the forecasted state fits the ground truth state in P times of all cases. To assess the reliability of the forecasts regarding the states, we use the decomposition of the Brier score [22] into reliability (REL_c), resolution (RES_c), and uncertainty (UNC_c) for each state c:

$$\mathrm{BS}_c = \mathrm{REL}_c - \mathrm{RES}_c + \mathrm{UNC}_c. \tag{5}$$

More details on the decomposition and the terms involved can be found in [22].

Regarding the evaluation of the QR forecast we evaluate whether the estimated frequency of the predictive distribution matches the observed frequency. We do this by creating Q-Q-plots (see [5]), i.e., we calculate how many times the true value is below a particular quantile divide by the total number of particular values. Additionally, we consider the continuous ranked probability score (CRPS) to assess the quality of the issued predictive time-to-transition distributions [17]. The CRPS for a deterministic forecast corresponds to the absolute error, i.e., mean CRPS (mCPRS) and mean absolute Error (MAE), respectively. This property of the CRPS allows us to compare both motion state forecasting methods (as detailed in Section V-C).

V. Experimental results

In this section, we present our experimental results. For evaluation, we use the data set described in Section IV-A. We test our approaches on 20 % of the cyclists, which are split in advance. The remaining 80 % of cyclists are used for training and hyperparameter optimization. For the latter, we split 20 % of the training data set.

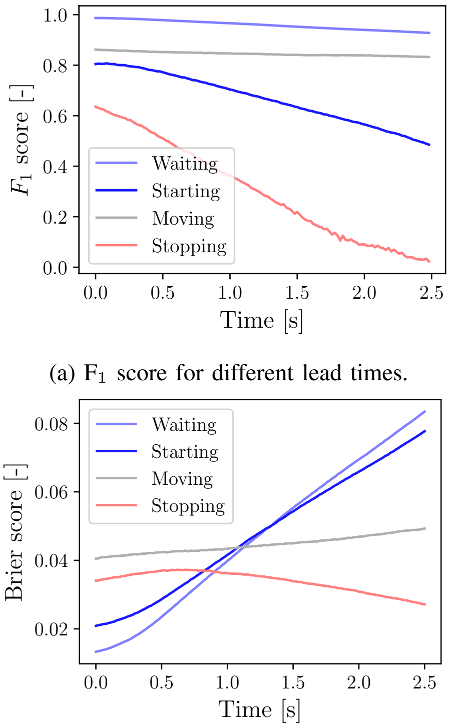
We use features derived from a sliding window segmentation of the cyclists' past head trajectory as features for both approaches. We use a sliding window length of 2 s and a sliding window step size of 20 ms. We approximate the trajectory in the sliding window using a polynomial approximation with orthogonal basis functions. The orthogonal expansion coefficients of the polynomial approximation are, in a least-squares sense, the best estimators of mean, slope, curvature of the approximated trajectory. We use these orthogonal expansion coefficients as features for our neural networks in both approaches. We refer to [2] for a detailed description.

A. Motion State Forecasting using Discrete Time Steps

In this section, we detail the experimental findings regarding motion state forecasting using discrete time steps. We perform a coarse grid search to optimize the neural network's hyperparameters, e.g., the number of hidden layers, and learning rate. We use the Brier score as our optimization criterion.

The best neural network found has two hidden layers with 2500 and 100 neurons, the hyperbolic tangent as activation function and uses for training a batch size of 1000 sliding window segments, and a learning rate of $0.5 \cdot 10^{-5}$.

The results of the approach are depicted in Fig. 6a. We see that the F_1 score deteriorates with increasing lead time. This is particularly evident for the motion states *Starting* and *Stopping*. This effect is probably due to the large class imbalance in the data set. *Waiting* and *Moving* together comprise about 85 % of the data set. Moreover, *Waiting* and *Moving* are long coherent segments. This is reflected in the prediction of the neural network, e.g., if the cyclist is *Waiting* and there are no current indicators for *Starting* the network's prediction is most likely *Waiting* which is correct in most of the cases.



(a) F_1 score for different lead times.

(b) Brier score for different lead times and all four motion states.

Fig. 6: Results in terms of F_1 (higher is better) and Brier score (lower is better) for motion state forecasting using discrete time steps.

The Brier score deteriorates for higher lead times, especially for *Waiting* and *Starting*. The Brier score for *Moving* and *Stopping* remains approximately constant. *Stopping* is rarely correctly predicted for larger time horizons. However, the models predict *Stopping* with a relatively small probability, leading to a slightly improved Brier score due to the under-representation of the *Stopping* class. We depict the reliability of the model's predictions in terms of the Brier score decomposition (cf. Eq. 5) for different lead times in Tab. I. We see the model is well-calibrated, even for larger lead times.

In Fig. 7, we depict a motion state forecast for a cyclist about to transition from *Stopping* to *Waiting*. We see that the transition from *Stopping* to *Waiting* is detected well.

REL_c BS_c	0.0 s	0.5 s	1.0 s	1.5 s	2.0 s	2.5 s
Waiting	0.0001 0.013	0.0003 0.024	0.0007 0.040	0.001 0.055	0.0005 0.069	0.0003 0.083
Starting	0.0001 0.021	0.0002 0.029	0.0007 0.042	0.0010 0.054	0.0006 0.066	0.0004 0.078
Moving	0.0005 0.041	0.0005 0.042	0.0004 0.043	0.0003 0.045	0.0003 0.047	0.0003 0.049
Stopping	0.0002 0.034	0.0001 0.037	0.0002 0.036	0.000007 0.033	0.0001 0.031	0.0001 0.027

TABLE I: Reliability (REL_c) and Brier (BS_c) score for different lead times.

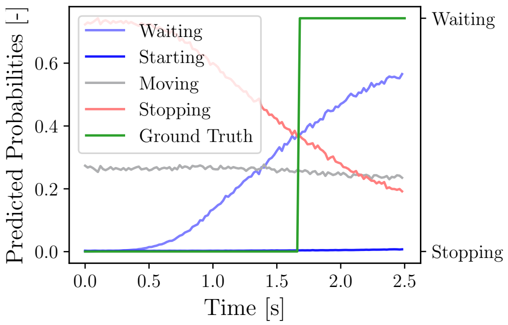


Fig. 7: Example motion state forecast for transition from *Stopping* to *Moving* using discrete time steps approach. The ground truth motion state is depicted in green, i.e., right axis of the plot.

B. Motion State and Time-to-Transition Forecasting

In this section, we review our experimental findings regarding our novel two-stage motion state forecasting methodology. Again, we optimize the hyperparameters of the neural networks involved using a coarse grid search. The best network for classifying the current and next motion state (i.e., the motion state prediction network) has two hidden layers with 500 and 50 neurons. It is trained using a batch size of 1000 and a learning rate of $0.5 \cdot 10^{-6}$. As before, we observe that the current motion state is better predicted than the next motion state (cf. Tab. II). Moreover, we also see from the Brier score's decomposition that the predictions are calibrated, i.e., reliable.

For the QRNN (i.e., the time-to-transition network), we use the following hyperparameters: One hidden layer with 100 neurons and rectified linear activation functions. The network is trained with a batch size of 1000 and a learning rate of 10^{-6}. We consider the quantile levels 1 %, 5 %, 10 %, 20 % …, 90 %, 95 %, and 99 %. We perform a reliability analysis to see whether the predictive distribution is properly calibrated. The results of this investigation are depicted in the Q-Q-plot in Fig. 8. We see that the issued predictive distributions are well-calibrated.

The QRNN predicts the time to transition well in the short term; however, the predicted distributions have a large variance for longer time horizons. In Fig. 9, we depict an example QRNN forecast for a transition from *Stopping* to *Waiting*. From the cumulative density function's (CDF) slope, we can see that the predictive distribution is very dense at the ground truth transition.

	State	Waiting	Starting	Moving	Stopping
BS_c	current next	0.0127 0.0832	0.0196 0.0793	0.0407 0.0511	0.0344 0.0294
REL_c	current next	0.000111 0.000437	0.000167 0.000590	0.000331 0.000415	0.000125 0.001137
F_1	current next	0.988 0.926	0.814 0.487	0.861 0.826	0.648 0.118

TABLE II: Brier (BS_c) score, the Reliability (REL_c), and the F_1 for each motion state of the motion state and time-to-transition forecasting approach.

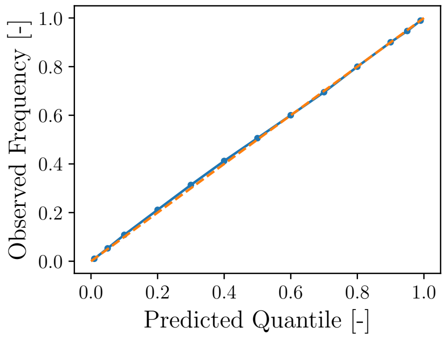


Fig. 8: Results of reliability analysis, i.e., Q-Q-plot of time-to-transition network. The blue dotted line depicts the calibration of the QRNN model and the orange dashed line corresponds to a perfectly calibrated model.

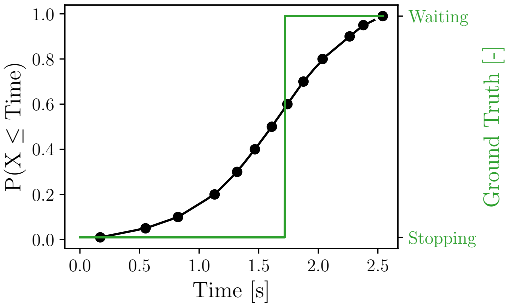


Fig. 9: Sample scene of cyclist about to transition from the *Stopping* to *Waiting*. The CDF of the QRNN's prediction is given by the black dotted line with the dots being the quantile levels. The green line, i.e., right axis of the plot, denotes the true motion state.

C. Comparison of Approaches

In this section, we compare both approaches to cyclist motion state forecasting. For this purpose, we first look at the ability of the two approaches to forecast transitions and, in the second step, we examine the quality of the predicted time to transitions. The results of the first analysis, i.e., the approaches' ability to detect motion state transitions within the next 2.5 s, are depicted in Fig 10. We use confusion matrices, also referred to as transition detection matrices, for this purpose. We see that both approaches perform approximately equally well. However, still in many cases, the transition is not detected, although there is a transition within the considered forecasting horizon. This confirms the investigations of Hubert et al. in [11] that many cyclists change their state (e.g., start to move) without any sign being recognizable in advance. Only very shortly before the actual starting a prediction about the change of the state of motion can be made based on the past movement.

The comparison of the two approaches with respect to the time to transitions is more difficult. The time to transitions of the first approach extracted from the class probabilities are deterministic, and the predictions of the QRNN are probabilistic. Therefore, in the first step, we interpret the QRNN prediction as a deterministic forecast (i.e., we consider the 50 % quantile). In the second step, we examine the MAE and the mean CRPS to compare the approaches. The distribution of the residuals of the two approaches are shown in Fig. 11.

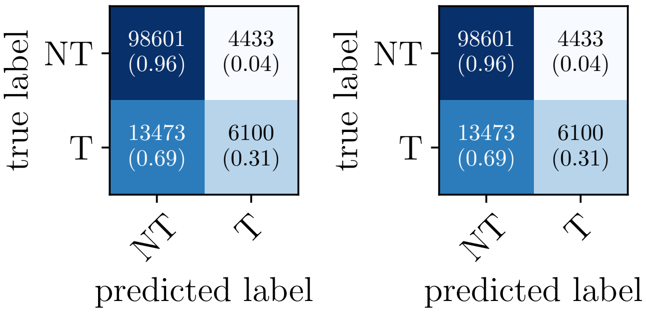
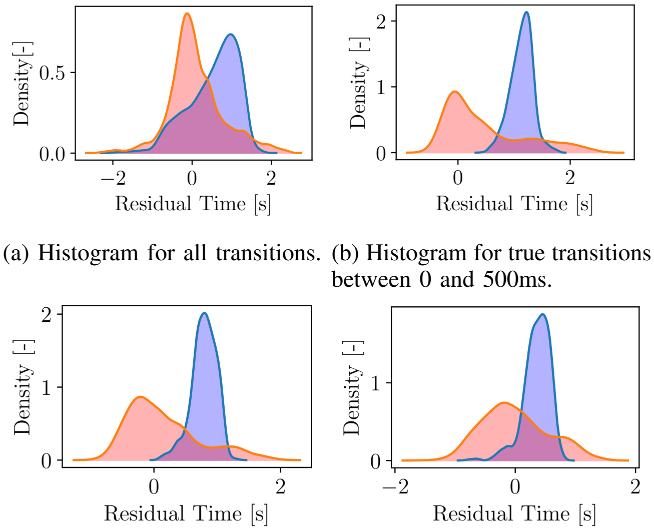


Fig. 10: Transition detection matrices for the motion state forecasting using discrete time steps (on the left) and motion state and time-to-transition forecasting (on the right) with T and NT denoting *Transition* and *No Transition*, respectively.

We see that the time to transition derived from the discrete time step approach is relatively unbiased, even for higher lead times. However, we see that a second maximum is created, which is around 1.25 s, i.e., half of the maximum forecast horizon. This suggests that a kind of averaging is taking place here, i.e., several motion states become similarly likely for higher lead times. This is probably due to the fact that larger time horizons are difficult to predict based on past trajectories. Regarding 50 % quantile, the QRNN tends to overestimate the time to transition, as it can be seen in the residuals. This holds true for all considered time intervals.



(a) Histogram for all transitions.

(b) Histogram for true transitions between 0 and 500ms.

(c) Histogram for true transitions between 500 and 1000ms.

(d) Histogram for true transitions between 1000 and 1500ms

Fig. 11: Histograms of time-to-transitions residuals (predicted - true time). The histograms are smoothed by a kernel density estimation with a Gaussian kernel. The red and blue curves correspond to of the discrete time steps approach and QRNN, respectively. The density is scaled by 10^3.

	Waiting to Starting	Starting to Moving	Moving to Stopping	Stopping to Waiting
Discrete Time Steps [s] (MAE)	198,10	1380,00	420,00	269,10
QRNN [s] (mCRPS)	561.40	343.33	350.39	410.03

TABLE III: Mean absolut error and mean CRPS (mCRPS).

This might indicate that the QRNN predictions are reliable but rather vague and less specific. The spread of the first approach is much higher than the spread of predictions of the QRNN. The MAE and mean CRPS for different motion state transitions are depicted in Tab. III. This gives a slight intuition which of both methods predicts better for what motion state transitions. From the two studies, we can conclude that the discrete time step approach performs better on average, although it sometimes overestimates by a large margin. The probabilistic method tends to be vaguer but does not always perform as well.

VI. Conclusions and Future Work

In this article, we present two new approaches that go beyond motion state detection. These two approaches forecast cyclists' future motion states in real-world traffic scenarios using the cyclist's past trajectory. The first method forecasts motion states for discrete time steps over a forecast horizon of 2.5 s. The second method detects the current and the next motion state and estimates the time between the two using a probabilistic forecasting method. Both approaches can detect upcoming motion state transition reliably. However, both methods struggle to anticipate motion state transitions which are further in the future. This result confirms prior empirical investigations on early motion transition indicators [11]. Though for short forecast horizons, the methods can forecast movement transitions using the past trajectory reliably. In particular, the discrete time steps method has proven to be the best choice, as it has a lower mean time to transition forecasting error. The probabilistic method using QRNN performs slightly worse; however, it can also express time to transition prediction uncertainties. Moreover, it is not limited to the number of discrete forecasting time steps and can predict arbitrarily long prediction time horizons.

Our future work will focus on using these motion state forecasts as additional information to improve cyclist trajectory forecasting. In this context and to improve the forecasts, we will also investigate the usage of more expressive features such as features derived from camera images [25] or body poses [16]. Furthermore, we plan to extend our approach to forecasting the cyclist's lateral state machine, allowing us to forecast turning motions.

Acknowledgment

This work results from the project DeCoInt[2], supported by the German Research Foundation (DFG) within the priority program SPP 1835: "Kooperativ interagierende Automobile", grant numbers SI 674/11-2 and DO 1186/2 and the project KI Data Tooling (19A20001O) funded by the German Federal Ministry for Economic Affairs and Energy (BMWI).

References

[1] VRU Trajectory Dataset, https://www.th-ab.de/vru-trajectory-dataset, last downloaded 2021/03/30

[2] Bieshaar, M.: Cooperative Intention Detection using Machine Learning–Advanced Cyclist Protection in the Context of Automated Driving. Intelligent Embedded Systems, Kassel University Press (2021), (Dissertation, University of Kassel, Department Electrical Engineering/ Computer Science)

[3] Brier, G.W.: Verifcation of Forecasts Expressed in Terms of Probability. Monthly Weather Review **78**(1), 1–3 (1950)

[4] Cannon, A.J.: Quantile regression neural networks: Implementation in r and application to precipitation downscaling. Computers & Geosciences **37**(9), 1277–1284 (2011)

[5] Chambers, J.: Graphical methods for data analysis. Chapman & Hall statistics series, Wadsworth International Group (1983)

[6] E. A. I. Pool, E., Kooij, J.F.P., Gavrila, D.M.: Using road topology to improve cyclist path prediction. In: IV. pp. 289–296. Redondo Beach, CA (2017)

[7] European Road Safety Observatory: Annual Accident Report 2018 (2018), https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/pdf/statistics/dacota/asr2018.pdf, last accessed: 2021/03/30

[8] Fang, Z., Vázquez, D., López, A.: On-board detection of pedestrian intentions. Sensors **17**(10), 2193 (2017)

[9] Gneiting, T., Katzfuss, M.: Probabilistic forecasting. Annual Review of Statistics and Its Application **1**(1), 125–151 (2014)

[10] Goldhammer, M., Köhler, S., Zernetsch, S., Doll, K., Sick, B., Dietmayer, K.: Intentions of Vulnerable Road Users–Detection and Forecasting by Means of Machine Learning. T-ITS pp. 1–11 (2019)

[11] Hubert, A., Zernetsch, S., Doll, K., Sick, B.: Cyclists' starting behavior at intersections. In: IV. pp. 1071–1077. Redondo Beach, CA (2017)

[12] Keller, C.G., Gavrila, D.M.: Will the Pedestrian Cross? A Study on Pedestrian Path Prediction. T-ITS **15**(2), 494–506 (2014)

[13] Koenker, R., Hallock, K.: Quantile regression: An introduction. Journal of Economic Perspectives **15**(4), 43–56 (2001)

[14] Köhler, S., Goldhammer, M., Bauer, S., Zecha, S., Doll, K., Brunsmann, U., Dietmayer, K.: Stationary Detection of the Pedestrian's Intention at Intersections. ITS Magazine **5**(4), 87–99 (2013)

[15] Kooij, J.F.P., Flohr, F., Pool, E.A.I., Gavrila, D.M.: Context-based path prediction for targets with switching dynamics. International Journal of Computer Vision **127**(3), 239–262 (2019)

[16] Kress, V., Jung, J., Zernetsch, S., Doll, K., Sick, B.: Pose Based Start Intention Detection of Cyclists. In: ITSC. pp. 2381–2386. Auckland, New Zealand (2019)

[17] Matheson, J., Winkler, R.: Scoring rules for continuous probability distributions. Mangement Science **22**(10), 1–40 (1976)

[18] Quintero, R., Parra, I., Lorenzo, J., Fernandez-Llorca, D., Sotelo, M.A.: Pedestrian intention recognition by means of a hidden markov model and body language. In: ITSC. pp. 1–7. Yokohama (2017)

[19] Quintero Mínguez, R., Parra Alonso, I., Fernández-Llorca, D., Sotelo, M.A.: Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition. T-ITS **20**(5), 1803–1814 (2019)

[20] Saleh, K., Hossny, M., Nahavandi, S.: Early intent prediction of vulnerable road users from visual attributes using multi-task learning network. In: Conference on Systems, Man, and Cybernetics (SMC). pp. 3367–3372 (2017)

[21] Schreiber, J., Sick, B.: Emerging relation network and task embedding for multi-task regression problems. In: ICPR. Milan, Italy (2020)

[22] Siegert, S.: Variance estimation for Brier Score decomposition. Quarterly Journal of the Royal Meteorological Society **140**(682), 1771–1777 (2014)

[23] Statistisches Bundesamt: Verkehrsunfälle Kraftrad- und Fahrradunfälle im Straßenverkehr 2018 (2019)

[24] Zernetsch, S., Kohnen, S., Goldhammer, M., Doll, K., Sick, B.: Trajectory prediction of cyclists using a physical model and an artificial neural network. In: IV. pp. 833–838. Gotenburg, Sweden (2016)

[25] Zernetsch, S., Kress, V., Sick, B., Doll, K.: Early Start Intention Detection of Cyclists Using Motion History Images and a Deep Residual Network. In: IV. pp. 1–6. Changshu, China (2018)