2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

# Recurrent Neural Network Architectures for Vulnerable Road User Trajectory Prediction

Hui Xiong[1], Fabian B. Flohr[2], Sijia Wang[1], Baofeng Wang[3], Jianqiang Wang[1] and Keqiang Li[1]

***Abstract*— We present an experimental study comparing various Recurrent Neural Network architectures for the task of Vulnerable Road User (VRU) motion trajectory prediction in the intelligent vehicle domain. Making use of temporal motion cues and visual appearance features, we design multi-cue RNN-based architectures with dedicated optimization process to predict future moving trajectories from historical consecutive frames. Experiments are performed on image sequences recorded from on-board a moving vehicle and public tracking datasets. In particular, the *Tsinghua-Daimler Cyclist Benchmark (TDCB)* has been augmented with additional annotations (various VRU types) to support the evaluation of object tracking approaches and trajectory prediction methods. This newly introduced dataset is termed *TDCB-Track*. We demonstrate the effectiveness of the proposed RNN architectures on the public MOT16 dataset and the *TDCB-Track* dataset. We show that the proposed approaches outperform simpler baseline methods and stay ahead with the state-of-the-art.**

## I. INTRODUCTION

Forecasting the motion of surrounding objects is an important problem in many application domains, such as social robots, security surveillance, traffic management, sport analysis, and intelligent vehicles. Especially the prediction of the Vulnerable Road Users in the intelligent vehicle domain can be quite challenging due to their high maneuverability. A pedestrian for instance can change his motion in a while. Predicting the correct future positions of those VRUs, as presented in Fig. 1, is crucial for current Advanced Driver Assistant System (ADAS) and the future of self-driving cars.

Unlike the rapid development of vision-based VRU detection methods [2], comparatively little effort has been devoted to VRU motion tracking and trajectory prediction. Especially when object tracking and prediction involve the automotive domain with its complex urban traffic scenarios, variable weather, unknown number of objects as well as motion uncertainty of targets, this task can get quite challenging.

Traditional filtering methods such as the well-known Kalman Filter (KF) [1] or variants, Particle Filter (PF) [3] are designed to cope with linear or non-linear motions using the Bayesian formulation. Due to the applied explicit model assumptions sudden, heavier or unusual dynamic changes can often not be captured by those methods. Object motion trajectory prediction could be seen as a sequential learning task [4], [5]. Different to convolutional neural networks (CNNs) mainly used for feature extraction [6], RNNs are designed for sequence processing [4]. While conventional RNNs perform poorly for long sequence learning due to the vanishing gradient problem, Long Short-term Memory (LSTM) [5], [7] can better deal with the problem. However, current approaches utilizing LSTMs are not appropriate to be used for on-board multi-class VRU motion prediction. They either focus on motion temporal cues and ignore appearance features [4], or they suffer from high-dimensional computational complexity by taking the whole image information into account [16].

This paper addresses this by proposing various multi-cue LSTM-based architectures for VRU motion prediction, considering motion information, class labels and deep and locally extracted visual features for predicting the future trajectories of VRUs. We investigate in parameter optimization of different LSTM variants using different prediction cues, different historical time steps and prediction horizons, evaluated on the MOT16 and TDCB-Track dataset.

## II. PREVIOUS WORK

Multiple object tracking can be divided into three parts [4], [6], [8], [9]: object detection, motion prediction and data association. Our work focuses on the motion prediction for VRU trajectory prediction in an intelligent vehicle application. In this section, we discuss related work focusing on available datasets and existing approaches on motion

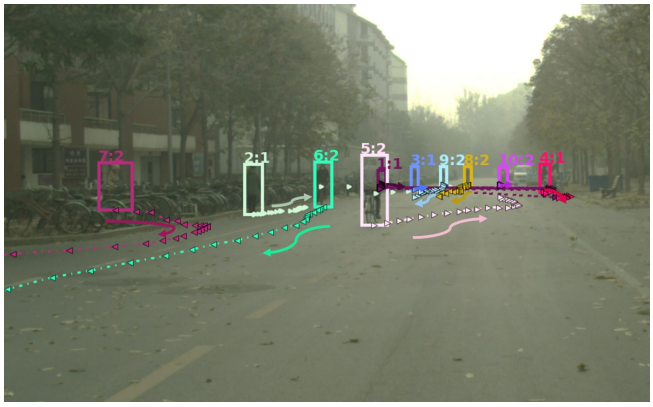



Fig. 1. Our method predicts VRU motion trajectories (as showed in the dash dot point) based on historical motion patterns which are learned from ground truth data. Different colors represent various objects. The text at the top of the bounding box (BB) is formatted as track id: class id (1 for pedestrians, 2 for riders), and the arrow indicates the moving direction.

Research supported by the National Science Fund for Distinguished Young Scholars (51625503) and the Major Project (61790561), and in part by Tsinghua University-Daimler Joint Research

[1]Hui Xiong, Sijia Wang, Jianqiang Wang and Keqiang Li (the corresponding author) is with the State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing, 100084, China

[2]Fabian B. Flohr is with the Environment Perception Department, Daimler AG Research and Development, 89081 Ulm, Germany

[3]Baofeng Wang is with the Research and Development of Autonomous Driving & Safety, Daimler Greater China Ltd., Beijing, China

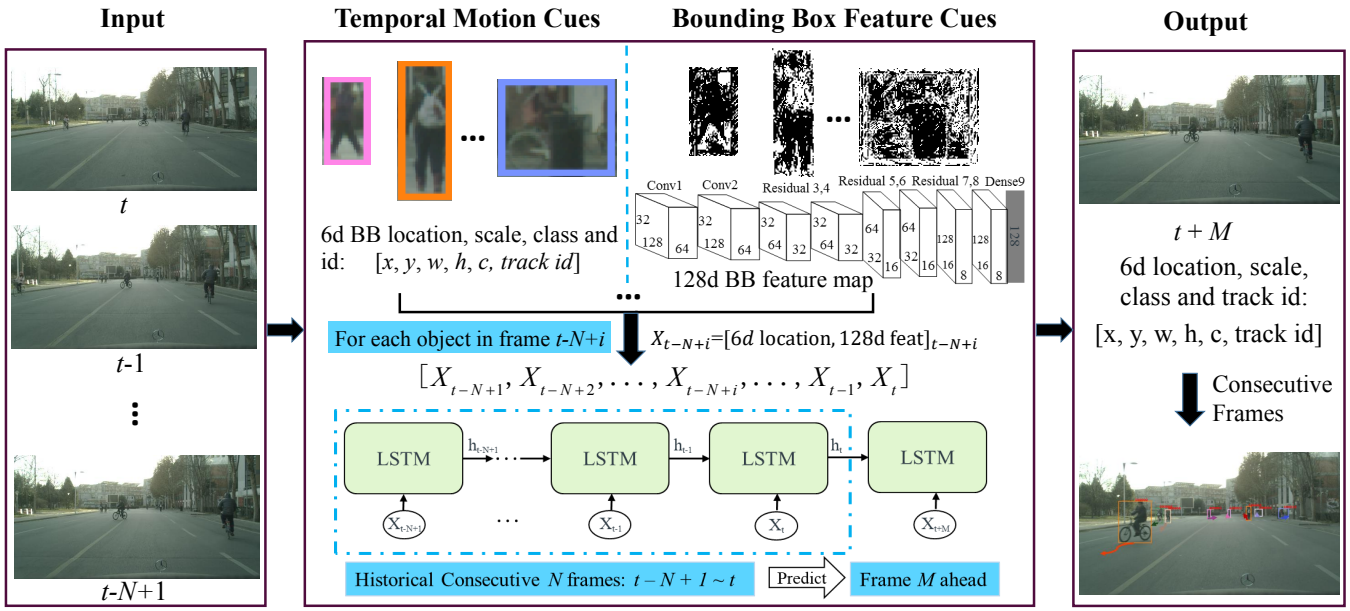



Fig. 2. An overview of our multi-target motion prediction framework based on LSTM. We use a separate LSTM network to predict the motion pattern for each object. Besides class type, bounding box position and size, the proposed method makes use of deep convolutional local features. At a current time step $t$, the model predicts $M$ frames ahead based on $N$ available historical frames (step size).

TABLE I

COMPARISON OF POPULAR MULTIPLE OBJECT TRACKING DATASETS TO OUR NEWLY INTRODUCED TDCB-TRACK DATASET BEING SPECIALLY DESIGNED FOR VRU TRACKING AND TRAJECTORY PREDICTION.

| Dataset | Camera (O=On-board, M=Moving, S=Static) | VRU Classes (P=Pedestrian, C=Cyclist, R=other Rider) | Train | | | Valid / Test | | | Total | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #seqs | Durations (mins) | #tracks | #seqs | Durations (mins) | #tracks | #seqs | Durations (mins) | #tracks |
| KITTI [10] | O | P+C | 21 | 13 | - | 0 / 29 | 0 / 18 | - | 50 | 30 | - |
| MOT15 [14] | S+M | P | 11 | 6 | 500 | 0 / 11 | 0 / 10 | 0 / 721 | 22 | 16 | 1,221 |
| MOT16 [11] | S+M | P | 7 | 4 | 517 | 0 / 7 | 0 / 4 | 0 / 759 | 14 | 8 | 1,276 |
| MOT17 [12] | S+M | P | 21 | 11 | 1,638 | 0 / 21 | 0 / 13 | 0 / 2,355 | 42 | 24 | 3,993 |
| PathTrack [13] | S+M | P | 640 | 161 | 15,380 | 0 / 80 | 0 / 11 | 0 / 907 | 720 | 172 | 16,287 |
| TDCB-Track (**ours**) | O | P+C+R | 9 | 65 | 11,117 | 2 / 2 | 4 / 10 | 142 / 478 | 13 | 79 | 1,737 |

trajectory prediction, including motion state models and trajectory prediction cues.

**Challenging tracking datasets.** KITTI [10], MOTChallenge [11], [12], [14] PathTrack [13] are publicly available for the evaluation of multiple object tracking approaches. See details in TABLE I, compared to the introduced TDCB-track dataset. Various datasets exist also for the task of single object tracking [16] including the popular OTB [27], the ETH human-trajectory dataset [34] and the UCY dataset [5]. These datasets have promoted technological progress in the task of visual tracking approaches. Among them, only KITTI [10] focuses on the intelligent vehicle domain and provides recorded sequences from an on-board camera mounted behind the windshield. The KITTI dataset concentrates on pedestrians and vehicles and includes only a limited amount of cyclist instances. PathTrack [13] provides various sequences including multiple application domains. Being reproduced and rendered virtually, the sequences of the datasets proposed in [11], [12], [14] are served as standard benchmarks for Multiple Object Tracking. Compared to these datasets, the introduced TDCB-Track exhibits on-board recorded sequences considering multiple VRU types including pedestrians, cyclists and other riders. Commonly used metrics for the evaluation of motion trajectory prediction is the Average Displacement Error [34] and the Final Displacement Error of the ground point [5], [21] or the bottom center of bounding box in meters [18], others include the center location error of the bounding box (computed as mean squared error) [22] and the bounding box overlap ratio, measured as an average overlap score (AOS) in pixels [16].

**Various motion state models exist.** The KF and PF are dominant approaches for motion state estimation using the Bayesian formulation [3]. The former is an efficient inference algorithm for linear dynamical system. With the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) [15] variants for non-linear motion and measurement system exist. The UKF estimates the latent state variables of non-linear system based on a sequential Monte Carlo importance sampling on observations. The Interacting Multiple Model (IMM) [19] filter implements a Switching Linear Dynamic System (SLDS) [32]) which can account for different motion models. Recently, the work of Kooij et. al. [33]) showed how to improve the SLDS formulation by incorporating explicit behavior cues of the VRU (i.e. head

orientation [31]). Data-driven models such as RNNs and their variants including LSTM [7] and GRU [17] provide some valuable approaches to estimate motion states directly based on visual cues from historical observations. Alexandre el al. [5] uses a separate LSTM for each trajectory, and proposes a pooling layer to jointly learn the interaction between each pedestrian in surveillance sequences. Tunmas et al. [26] derives a discriminative, deterministic state estimate method called Backprop KF to draw a connection between KF and LSTM evaluated on the KITTI dataset. The work of Dan et al. [22] trains two dependent LSTMs to output a predicted value and a Gaussian distribution for a single target motion.

**Trajectory prediction cues.** Motion cues such as the location, scale or speed are widely used in Kalman Filters [6], [9], RNN-based [4], LSTM-based [5], [21], [22] or mixed [25], [26] motion models. These approaches tend to be not effective in unconstrained traffic environments due to a loss of fine-grained visual appearances. To deal with the problem, recent works [16], [23], [24] integrated spatial visual appearances into the prediction model, demonstrating the power of RNN-based motion models to learn the dependencies between spatial image correlations. In particular, Wang et al. [23] uses CNN features from three RGB channels as measurements for stacked Convolutional LSTM. However, the approach concentrates on single object tracking using the whole image feature space as the input vector, which generates high-dimensional observations. Besides these RNN models, Generative Adversarial Networks [28] are employed to preserve the sharpness of the predicted frame in [29].

In summary, existing motion prediction methods are mainly applied to pedestrians [11]–[14] in static / moving perspective or to vehicles [5] in the automotive domain. The limited amount of cyclists or other riders in available dataset is a barrier for an application of multi-class VRU motion prediction approach. Existing motion state prediction methods are prone to focus on motion temporal cues [22], [25], learned by probabilistic linear / non-linear model [3], [15], [19] or data-driven RNN-based networks [5], [21]. Due to the rapid development of deep learning [6], [13], methods are able to integrate deep spatial appearance feature with temporal cues. However, most of these methods focus on single object tracking and are learned on the whole image [23], [24], coming with high computational costs which can be crucial for ADAS. Finally, we note that an extensive comparison of different RNN-based models variants for motion trajectory prediction is not available.

To address above shortcomings the contributions of this paper are three-fold: 1) variants of multi-cue data-driven motion prediction using RNN models are presented and applied in the intelligent vehicle domain, considering motion information, class labels and bounding box deep visual image features as trajectory prediction cues; 2) we carefully evaluate the different RNN variants and parameter optimization strategies using different data cues, different number of historical time steps and different prediction horizons on various datasets; 3) due to the lack of available automotive tracking and prediction datasets capturing various VRU types, we introduce the TDCB-Track dataset extending the dataset originally proposed in [2] as our third contribution.

## III. PREDICTION MODEL

We are interested in predicting the future motion of surrounding VRUs in image space. Our prediction model considers bounding box location, size and scale, VRU class type as well as deep visual image features.

### A. System Overview

The overview of the motion prediction is illustrated in Fig. 2. We use a separate data-driven LSTM network for studying the motion pattern of each object using motion cues and deep visual appearance feature cues from multiple historical consecutive frames. The proposed motion prediction model is a deep neural network that tasks as input of multiple object in raw video frames from GT returning the bounding box coordinates $[x, y, w, h]$, class id ($c$), track ID ($id$) of a tracked object (used to identify the same object in consecutive $N$ frames). Making used of an independence assumption condition between frames, we formulate the on-line motion prediction problem as sequential processing in LSTMs, factorizing the full object motion prediction probability into:

$$p(B_{motion}^{1:T,j}; B_{feat}^{1:T,j} | F^{1:T,j}) = \prod_{t=1}^{T} p(B_{motion}^{t,j} | B_{motion}^{<t,j}, B_{feat}^{<t,j}, F^{\leq t,j}) \quad (1)$$

where $B_{motion}^{t,j}$ and $B_{feat}^{t,j}$ are the 6d location and 128d deep visual image feature vector pooled from the bounding box of an object $j$ at frame $t$, $F^{t,j}$ is input raw frame *t* with object *j*. $B_{*}^{<t,j}$ includes all previous historical location or feature maps before frame time *t*, and $F^{\leq t,j}$ is the history of input frames up to frame time *t*.

Concretely, the LSTM-based prediction uses the $j$-th object information from historical *N* frames $[X_{t-N+1}, X_{t-N+2}, \cdot \cdot \cdot, X_t]$ as input, for each frame, the track $j$ has temporal motion and visual feature cues, expressed as $X_{t,j} = [BB_{motion}^{t,j}, BB_{feat}^{t,j}]$. The output is the predicted bounding box motion information in frame *M* ahead $\hat{X}_{t+M} = BB_{motion}^{t+M,j}$, also denoted by $BB_{pred}$. Among $[X_{t-N+1}, X_{t-N+2}, \cdot \cdot \cdot, X_t]$, *t* is the current frame index, *N* is the number of historical learned frames (step size), and *M* is the prediction horizon, i.e. the number of frames predicted ahead. The different proposed models have different input specifications which are summarized in TABLE II. The bounding box motion cues $BB = [x, y, w, h, c]$, defined normalized image coordinates, i.e. the top left corner and width and height of the bounding box. Normalized image coordinates are important for an easier regression incorporating also the L2 normalized 128d feature vector [20]. And finally, $c$ indicates the class id, 1 for pedestrians and 2 for riders. Riders include cyclist, motorcyclist, tricyclist, wheelchair user and moped rider, including the vehicle respectively, i.e. bike, motorbike, tricycle, wheelchair and moped.

TABLE II

$X_*$ PATTERNS IN RNN MODELS WITH VARIOUS PREDICTION CUES: TEMPORAL MOTION CUES (CONSIDERING PEDESTRIANS AND RIDERS JOINTLY OR INDIVIDUALLY), TEMPORAL MOTION WITH IMAGE FEATURE CUES (MOTION+IMGFEAT), OR WITH BB FEATURE CUES (MOTION+BBFEAT).

| Input | Temporal motion cues | | | motion+imgfeat cues | motion+bbfeat cues |
|---|---|---|---|---|---|
| | motion-hybrid | motion-pedestrian | motion-rider | | |
| $X_*$ | $[x, y, w, h, c]$ | $[x, y, w, h, 1]$ | $[x, y, w, h, 2]$ | $[x, y, w, h, c, 4096d$ image feature map$]$ | $[x, y, w, h, c, 128d$ bounding box feature map$]$ |

### B. Temporal Motion Cues for Prediction

Different classes of VRUs have different maneuverable attributes. Therefore, we design motion cues of VRUs refer to location, scale and class category, which can be acquired from from a state-of-the-art object detectors. We evaluate the difference in a separate or joint consideration of pedestrian and rider motion cues (i.e. independent pedestrian motion pattern, independent rider motion and hybrid pedestrian-rider motion, termed as "motion-pedestrian", "motion-rider" and "motion-hybrid"). Based on these motion patterns, various RNN/LSTM/GRU variants [7], [17] are designed for the experimental practices.

In our objective module, the Mean Squared Error (MSE) [16] is used for training:

$$L_{MSE} = \frac{1}{k}\sum_{i=1}^{k}(BB_{gt} - BB_{pred})^2 \tag{2}$$

where $k$ indicates the number of training sample, $BB_{gt}$ is the target ground truth $[x, y, w, h]$ ignoring class category $c$, and $BB_{pred}$ is the model's prediction.

In TABLE II different cues are presented including *motion+imgfeat* cues and *motion+bbfeat* cues describing temporal motion with spatial image feature cues and temporal motion with spatial bounding box feature cues respectively, as input for LSTM-based motion prediction models in the following two parts in this section.

### C. Bounding Box Feature Map Cues for Prediction

To extend the RNN learning and analysis into the spatiotemporal domain, the proposed multi-cue motion prediction model makes use of temporal motion and spatial appearance information including location $[x, y]$, scale $[w, h]$, and class category $[c]$, as well as an extracted local feature map (i.e. a 128d feature vector) based on the bounding box. Different from the deep appearance descriptors used often for data association metrics [6], we regard the deep visual image appearance features as an important cue for RNN-based motion prediction.

The feature extractor is trained on a re-identification dataset [20], being originally applied for learning a deep association metric for multi-target tracking. The 128-dimensional feature vector is extracted from the 10-th dense layer, projected onto the unit hypersphere by using the final batch and L2 normalization [6] (two convolutional layers and six residual blocks, followed by L2 normalization). It is known that the object detector has an influence on the predicted results in different levels. For fair comparison and a deep understanding of motion prediction itself, ground truth is provided as detected bounding box with accurate track id and class id information. And track id is employed to identify the same object during historical consecutive $N$ frames.

### D. Image Feature Map Cues for Prediction

Due to the size of detected VRU bounding box ought not to be fixed and also the whole image sparse feature map could introduce the contextual information. Therefore we employ motion and image feature map cues [6] for prediction as a contrast experiment with the proposed motion and bounding box feature map cues-based RNN for prediction.

Besides the cues of location $[x, y]$, scale $[w, h]$, and class category $[c]$, we adopt regression-based YOLO object detection model and extract feature from the entire image instead of local bounding box feature extraction. Due to the fact that the YOLO network has global reasoning about the image during the prediction phase, encoding contextual information about appearance and classes, also has good generalization ability. Specifically, YOLO object detection network has 24 convolutional layers followed by 2 fully connected layers. The 4096d fully connected layer is used as the feature map.

## IV. EXPERIMENTS

In the following sections, we introduce applied datasets, the baselines and evaluation metrics for our experiments. Furthermore, we evaluate the prediction performance of different methods on the public MOT16 dataset [11] and our TDCB-Track.

### A. Datasets

**MOT16 dataset**. The dataset is widely used for presenting target tracking results focusing on pedestrians [11]. It contains seven training and seven test challenging video sequences in unconstrained environments filmed with both static and moving cameras with variable frame rates in $\{14, 25, 30\}$ fps. Since the annotations of the test set are unavailable, we separate one sequence (i.e. MOT16-02), as validation set from the seven training sequences.

**TDCB-Track dataset**. This dataset extends the TDCB [2] and has been collected from a moving vehicle with onboard automotive sensors. The images have been recorded with frame rate of 25 fps and an image resolution of $2048 \times 1024$ pixels. We extended the TDCB training set and have additionally labeled all VRUs (instead of only cyclists as in the original TDCB dataset). Each 10th frame has been annotated. TDCB validation and testing dataset are kept untouched, and are used here for validation and testing as well. For later experiments, we fine-tune hyper parameters on the validation set firstly, and then evaluate on test set.

The naïve way to annotate trajectories is to label the track in every frame. However, this approach becomes inefficient when the object velocity is slow and the frame rate is high, i.e. objects tend to move little between frames. Instead of a possible linear interpolation between annotated frames, we attempt to use a modification of LSTMs to learn data-driven motion model directly at the given labeling interval. In other words, each annotated frame we use in later TDCB-Track experiments covers the time length of ten collected frames, which is equal to 0.4 s. Although this time interval seems quite big to cover VRU dynamic changes, same time occupation has already been successfully applied in the Social LSTM approach [5].

### B. Evaluation Protocol

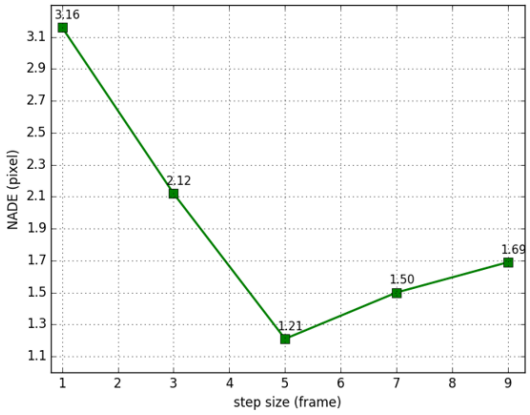
**Baselines.** During evaluation time, we design the following state-of-the-art methods [5], [18], [22] as multiple control setting covering varied linear/non-linear models:

- Linear model (LIN). We use a linear velocity model to extrapolate trajectories from center locations $[x, y]$ and sizes $[w, h]$ of a bounding box object. For a fair comparison, the number of historical frames (step size) is the same as with the other approaches.
- Kalman Filter (KF). We apply a linear dynamic system implemented by a KF applying a constant velocity model. Here, we represent the state variable $[x, y, w, h, v_x, v_y, v_w, v_h]$, instead of $[x, y, w, h]$ used in [22], where are the center location, width and height of bounding boxes (in pixels), and $v_x, v_y, v_w, v_h$ are the corresponding velocity of these observed variables (in pixels/frame). With the same step size ($N$) configuration as LSTM models, we predict the $(N+1)$th location and size learning from the historical $N$ locations and sizes of an object iteratively using KF. Process noise covariance matrix $Q$ is estimated by expectation maximization on all $N$ groups of training input data iteratively, and measurement noise $R$ is estimated as the covariance matrix of the difference between GT and predicted positions, initialized to 10.
- LSTM model with temporal motion cues [4], and / or combined with the holistic image feature cues (motion+imgfeat) [16]. An empirical exploration of different RNN variants (RNN/LSTM/GRU) variants and parameter configurations (peephole/layer normalization (ln)/Xavier initialization (xavier) [30]) is conducted. These baselines use the same parameter configuration as our newly proposed multi-cue LSTM-based motion prediction method.

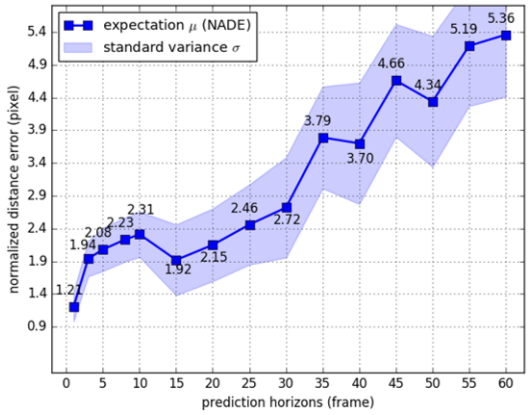
TABLE III

$5 \rightarrow 1$ MOTION PREDICTION COMPARISONS (NADE$\pm\sigma$) ON DIVIDED MOT16 VALIDATION DATASET – MOT16-02.

| Methods | | Metric (NADE) |
|---|---|---|
| LIN | | 5.72±3.89 |
| KF | | 2.33±0.59 |
| LSTM | Motion cues (Temporal motion cues) | 1.21±0.92 |
| | Motion+imgfeat cues | 3.87±2.01 |
| | Motion+bbfeat cues (**Ours**) | 1.10±1.06 |
| | Motion+bbfeat-ln cues (**Ours**) | 1.29±0.88 |
| | Motion+bbfeat-peephole cues (**Ours**) | 1.07±0.94 |
| | Motion+bbfeat-xavier cues (**Ours**) | **1.03**±0.88 |




(a) NADE for a prediction of one step ahead using an increasing number of historical frames (step size)




(b) NADE over increasing prediction horizons with a fixed number of historical frames, i.e. step size is 5

Fig. 3. Results on the MOT16 validation sequence (MOT16-02) using an LSTM-based model with temporal motion cues (termed as "LSTM_motion"). Note that only the pedestrian is considered on MOT16.

**Metrics.** Let time step $t = T + t_{future\_step}$ with $t_{future\_step}$ represent the predicted time step ahead. The predicted bounding box output of an object $j$ is $[\hat{x}, \hat{y}, \hat{w}, \hat{h}, \hat{c}]$, where $\hat{x}$ and $\hat{y}$ represents the top left corner point, $\hat{w}$ and $\hat{h}$ the width and height and $\hat{c}$ indicates the class of the predicted bounding box.

The bounding box center point $[\hat{x}_{bc}, \hat{y}_{bc}$ is gained by:

$$\begin{cases} \hat{x}_{bc} = \hat{x} + w/2 \\ \hat{y}_{bc} = \hat{y} + h \end{cases} \tag{3}$$

Formally, following a similar metric as described in [22], we design the normalized average displacement error (NADE) relative to the image size in the pixel level (lower is better):

$$\frac{\sum_{i=t+M}^{L} \sum_{j=1}^{K(i)} \sqrt{(x_{bc}^{i,j} - \hat{x}_{bc}^{i,j})^2 + (y_{bc}^{i,j} - \hat{y}_{bc}^{i,j})^2} / \sum_{i=t+M}^{L} K(i)}{\sqrt{w_{img}^2 + h_{img}^2}} * 100 \tag{4}$$

where $i$ denotes the frame index, $j$ the object index, $(w_{img}, h_{img})$ represents the width and height of the given image, $L$ is the sequence length, $M$ denotes the predicted frame step ahead and $K(i)$ indicates the number of VRUs in $i$-th frame. Normalization is necessary because we need to evaluate different datasets MOT16 and TDCB-Track with various image sizes under uniform standards.

Additionally an overlap score (intersection over union between predicted and ground-truth bounding box) is used. Specifically, we use the measured metric as average overlap score (AOS) in pixels [16] for multiple objects over all test sequences.

### C. Results and Analysis

**MOT16 dataset.** One epoch means one forward pass and one backward pass of all training samples. We first test the learning capability of the LSTM using temporal motion cues with varied number of historical frames (step size) $N$ and predicted frame steps ahead (prediction horizons) $M$ with 20 epochs, as illustrated in Fig. 3.

See TABLE III, we compare the proposed multi-cue motion prediction model with other motion prediction models based on temporal motion cues or motion+imgfeat cues under the optimal $N \rightarrow M$ combination ($N \rightarrow M$ means using $N$ historical frames to predict $M$ frames ahead). Moreover, we also analyze various LSTM-based optimization strategies including peephole, layer normalization, and Xavier initialization to serve as a reference of the best optimization strategy for proposed method on TDCB-Track.

Results in Fig. 3(a) show that LSTM has the best performance with step size 5 for one-step ahead. Fig. 3(b) illustrates how different prediction horizons using 5 historical frames affects the prediction performance.

A single object's trajectory on MOT16-02 sequence is shown in Fig. 4, proving the effectiveness of proposed method. Quantitative results in TABLE III support that the proposed LSTM architecture based on motion and bounding box feature cues is superior to other baselines (e.g. LIN, KF and LSTM-based with different cues). Peephole and Xavier initialization increase the performance, while layer normalization have a bad effect. Hence, Xavier weight initialization will be applied in our prediction method (named as *motion+bbfeat-xavier*) in the later evaluation on the TDCB-Track dataset.

**TDCB-Track dataset.** One training epoch means training prediction model on each sequence data of the 9 sequences from TDCB-Track training dataset (9741 frames). Training epoch and step size are fine-tuned using the GRU method (see Fig. 5). Note that $N \rightarrow 1$ means to observe a trajectory for $N \times 0.4s$ and predict paths at $1 \times 0.4s$ ahead.

Moreover, we test training loss of each sequence under various epochs on TDCB-Track training set, from which we observe that the training loss continue to fall, slowly after 20 epochs, when the training epoch is close to 200, the loss tends to remain unchanged. For comparison, we use 200 as maximum training epochs for experiments. From Fig. 5, we can see that 200 epochs with a step size equal to 3 is the optimal hyper-parametric combination for TDCB-Track dataset. In order to guarantee the convergence for deep and complex neural networks, we used Adam optimizer with a small learning rate of $10^{-5}$, which served as a basic parametric combination for RNN variants.

Different RNN/LSTM/GRU prediction variants are shown in TABLE IV. Each of the variants covers motion cues (includes separate motion for pedestrian and for rider), motion+bbfeat cues and motion+imgfeat cues. TABLE V shows also the comparison to baseline approaches (LIN and KF). For NADE, we outperform these simpler baselines by a large margin. For the AOS metric, we get the second best score. It is worth noting that the top two predictors are based on our designed GRU recurrent neural network. For one single track, it is shown in Fig. 6 that proposed method could

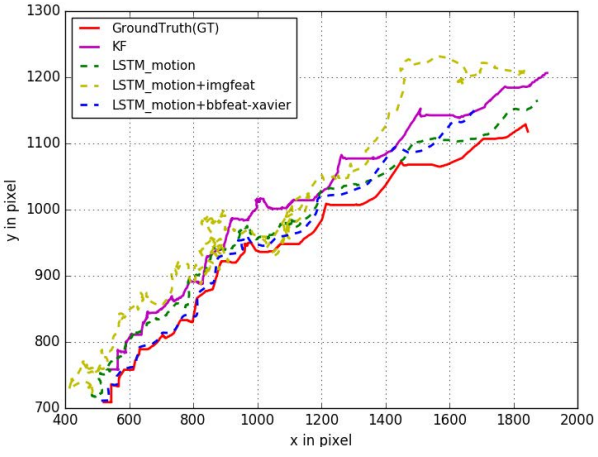



Fig. 4. The motion trajectory of a single object on MOT16-02 trained as 5→1 (i.e. five historical frames, predicted one frame ahead): KF (linear model) by purple solid lines, LSTM models with temporal or image feature cues by green or yellow dotted lines. Blue dotted and Red solid lines indicate our proposed one (**LSTM_motion+bbfeat-xavier**) and the ground truth, respectively.

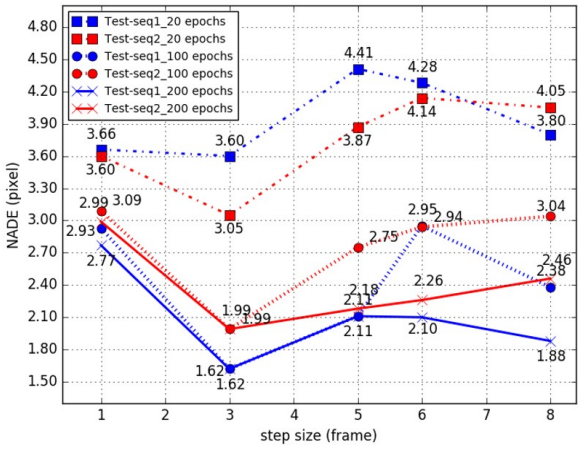



Fig. 5. NADE under various step sizes for a prediction of one step ahead and evaluated on the TDCB-Track set with 20/100/200 epochs (× 9741 iterations) on two test sequences. When the training epoch is close to 200, the loss tends to remain unchanged, showing that 200 epochs with step size 3 are the optimum parameters. Compared to $5 \rightarrow 1$ for MOT16, $3 \rightarrow 1$ is best combination for TDCB-Track.

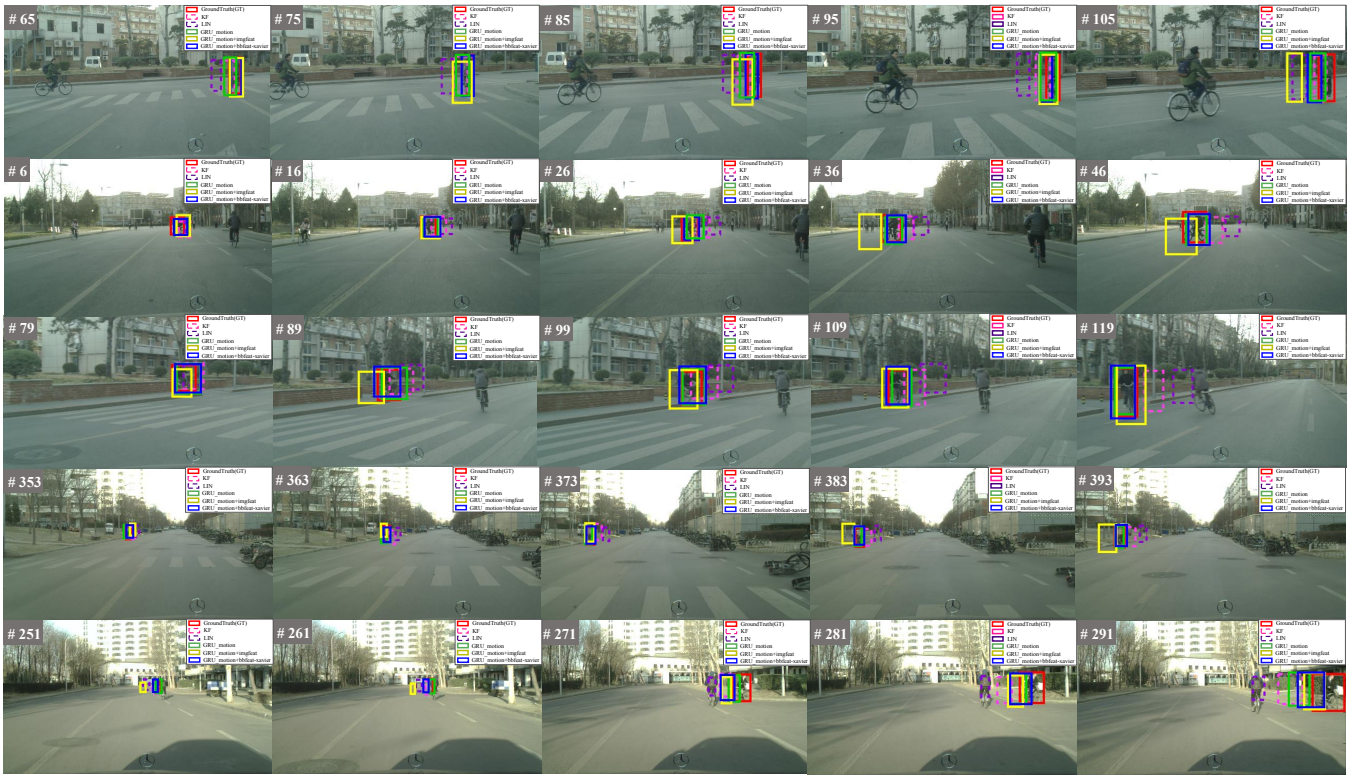


Fig. 6. Single object trajectory prediction on various motion scenes from TDCB-Track test set under various 3→1 motion prediction models. Dotted lines for LIN and KF models, the solid lines for GRU models, and Blue and Red bounding boxes indicate our method and GT, respectively. The first four rows represent good prediction result, the last row show bad prediction result, mainly coming from sudden object occlusion. In this case, deep visual feature cues have a negative effect on the prediction.

track some single tracks on motion dynamic change better than others on TDCB-Track test dataset both for common motion scenes and specific motion dynamic scenes. Further, in Fig. 7, we calculate the average tracking success rate over various overlap thresholds of multi-object bounding box prediction results to evaluate temporal robustness (the legend indicates method name and the average success rate). The result proves the effectiveness of proposed GRU-based multi-cue trajectory prediction method (3→1). Some results of proposed GRU-based motion prediction model with spatiotemporal multi-cues on the TDCB-Track dataset in campus scenarios with frequent moving change and occlusions with non-linear motion patterns are presented in Fig. 8.

## V. CONCLUSIONS

An experimental study comparing various Recurrent Neural Network architectures was presented for the task of Vulnerable Road User (VRU) motion trajectory prediction in the intelligent vehicle domain. Fully exploiting both temporal motion cues and visual appearance features, the applied

TABLE IV

MOTION PREDICTION COMPARISON OF DIFFERENT RNN VARIANT MODELS (3→1) ON TDCB-TRACK VALIDATION DATASET.

| Metric | Sequence | RNN Variant | Temporal motion cues based prediction | | | Motion+imgfeat cues based prediction | **Ours(Motion+bbfeat cues based prediction)** | |
|---|---|---|---|---|---|---|---|---|
| | | | motion-hybrid | motion-pedestrian | motion-rider | | motion+bbfeat | motion+bbfeat-xavier |
| NADE (pixel) | Valid set | RNN | 2.94 | 2.40 | 2.39 | 4.20 | 1.55 | **1.49** |
| NADE (pixel) | Valid set | LSTM | 2.02 | 2.13 | 2.46 | 3.51 | 1.46 | **1.40** |
| NADE (pixel) | Valid set | GRU | 1.80 | 2.25 | 2.37 | 3.21 | 1.30 | **1.24** |

TABLE V

PREDICTION PERFORMANCE (NADE AND AOS) ON TDCB-TRACK TEST DATASET FOR ALL THE METHODS (3→1). THE BEST AND SECOND BEST RESULTS ARE IN BOLD AND ITALIC FORMAT, ↓ MEANS LOWER IS BETTER AND ↑ MEANS HIGHER IS BETTER.

| Metric | Sequence | LIN | KF | GRU | | |
|---|---|---|---|---|---|---|
| | | | | Temporal motion prediction cues | Motion+imgfeat prediction cues | **Ours (Motion+bbfeat-xavier prediction cues)** |
| NADE $\pm\ \sigma\downarrow$ | Test set | 3.70 ± 4.86 | 1.78 ± *2.52* | *1.51* ± 3.82 | 3.16 ± 4.44 | **1.22 ± 2.41** |
| AOS ↑ | Test set | 0.151 | 0.321 | **0.424** | 0.175 | *0.400* |

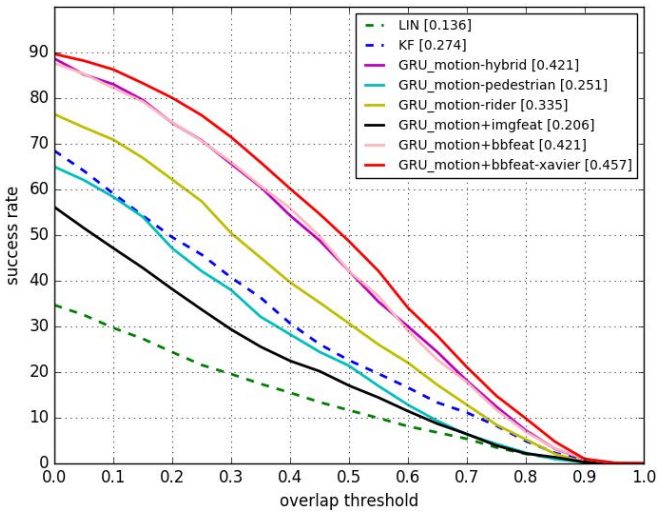



Fig. 7. Success rate over various thresholds for TDCB-Track (3→1): the dotted lines for LIN and KF models, the solid lines for GRU models with various cues. Pink and Red curve indicates our *motion+bbfeat* and *motion+bbfeat-xavier* in GRU version respectively.

Fig. 8. Accurate prediction results of our *motion+bbfeat-xavier* in GRU version (3→1) on TDCB-Track with various motion patterns. White '+' points: the lower center of GT, other '→' points: predicted lower center position, the same color as bounding box of each track. Track id and class id are shown on the top text of bounding box. To visualize the motion trajectory better, dotted dots are used to connect the ground-truth or predicted points.

networks learn to predict future moving trajectories from historical consecutive frames. Results have been gained on different datasets including the public available MOT16 and the newly introduced TDCB-Track dataset. The careful evaluations show that the GRU-based architecture with temporal motion and bounding box feature cues using Xavier weight initialization achieve the-state-of-art performance upon the evaluation on common AOS and NADE metrics. We could also show that it is important to tune the parameters using when using different dataset (e.g. the optimal number of historical frames is 5 for MOT16 and 3 for TDCB-Track). Our extensive experimental results and performance comparison with state-of-the-art motion prediction models demonstrate that LSTM-based model, especially GRU version with Xavier initialization, is powerful for sequential motion prediction task, and our predictor with spatiotemporal cues gets a better result.

## REFERENCES


[1] S. Lefèvre, D. Vasquez, and C. Laugier, A survey on motion prediction and risk assessment for intelligent vehicles, ROBOMECH Journal, vol. 1, no. 1, pp. 1-14, 2014.

[2] X. Li, et al., A new benchmark for vision-based cyclist detection, in IV, 2016, pp. 1028-1033.

[3] N. Schneider, D. M. Gavrila, Pedestrian path prediction with recursive Bayesian filters: A comparative study, in GCPR, 2013, pp. 174-183.

[4] A. Milan, et al., Online multi-target tracking using recurrent neural networks, in AAAI, 2017: 4225-4232.

[5] A. Alahi, et al., Social LSTM: Human trajectory prediction in crowded spaces, in CVPR, 2016.

[6] N. Wojke, A. Bewley, and D. Paulus, Simple online and realtime tracking with a deep association metric, in ICIP, 2017, pp. 3645-3649.

[7] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural computation, vol. 9, no. 8, 1997, pp. 1735-1780.

[8] Y. Xiang, A. Alexandre, and S. Silvio, Learning to track: Online multi-object tracking by decision making, in ICCV, 2015, pp. 4705-4713.

[9] A. Bewley, et al., Simple online and realtime tracking, in ICIP, 2016, pp. 3464-3468.

[10] A. Geiger, et al., Are we ready for autonomous driving? The KITTI vision benchmark suite, in CVPR, 2012, pp. 3354-3361.

[11] A. Milan, et al., MOT16: A benchmark for multi-object tracking, arXiv preprint, arXiv: 1603.00831, 2016.

[12] R. Henschel, et al., Fusion of head and full-body detectors for multi-object tracking, in CVPR Workshops, 2018, pp. 1428-1437.

[13] S. Manen, et al., Pathtrack: Fast trajectory annotation with path supervision, in ICCV, 2017, pp. 290-299.

[14] L. Leal-Taixé, et al., Motchallenge 2015: Towards a benchmark for multi-target tracking, arXiv preprint, arXiv: 1504.01942, 2015.

[15] J. Tao and R. Klette, Tracking of 2D or 3D irregular movement by a family of unscented Kalman filters, J. Inf. Convergence Commun. Eng., vol. 10, no. 3, pp. 307-314, Jul. 2012.

[16] G. Ning, et al., Spatially supervised recurrent convolutional neural networks for visual object tracking, in Circuits and Systems, 2017 IEEE International Symposium on, pp. 1-4.

[17] J. Chung, et al., Empirical evaluation of gated recurrent neural networks on sequence modeling,arXiv preprint, arXiv:1412.3555, 2014.

[18] C. G. Keller and D. M. Gavrila, Will the pedestrian cross? A study on pedestrian path prediction, PAMI, vol. 15, no. 22, 2014, pp. 494-506.

[19] Li, X. Rong and Vesselin P. Jilkov, Survey of maneuvering target tracking. Part I. Dynamic models, IEEE Transactions on aerospace and electronic systems, vol. 39, no. 4, pp. 1333-1364, 2003.

[20] L. Zheng, et al., MARS: A video benchmark for large-scale person re-identification, in ECCV, 2016.

[21] A. Robicquet, et al., Learning social etiquette: Human trajectory understanding in crowded scenes, in ECCV, 2016, pp. 549-565.

[22] D. Iter, et al., Target tracking with kalman filtering knn and lstms, 2016.

[23] Y. Wang, et al., Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms, In Advances in NIPS, 2017, pp. 879-888.

[24] E. Rehder, et al., Pedestrian prediction by planning using deep neural networks, in ICRA, 2018, pp. 1-5.

[25] C. Ju, Z. Wang and X. Zhang, Socially aware kalman neural networks for trajectory prediction, arXiv preprint, arXiv: 1809.05408, 2018.

[26] T. Haarnoja, et al., Backprop kf: Learning discriminative deterministic state estimators, in Advances in NIPS, 2016, pp. 4376-4384.

[27] Y. Wu, ey al., Object tracking benchmark, PAMI, 2015, vol. 37, no. 9, pp. 1834-1848.

[28] E. L. Denton, S. Chintala, and R. Fergus, Deep generative image models using a laplacian pyramid of adversarial networks, in Advances in NIPS, 2015, pp. 1486-1494.

[29] M. Mathieu, C. Couprie, and Y. LeCu, Deep multi-scale video prediction beyond mean square error, in ICRL, 2016.

[30] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in AISTATS, 2010, pp. 249-256.

[31] F. Flohr, et al., A probabilistic framework for joint pedestrian head and body orientation estimation, TITS, vol. 16, no. 4, 2015, pp. 1872-1882.

[32] J. F. P. Kooij, et al., Context-based pedestrian path prediction, in ECCV, 2014, pp. 618-633.

[33] J. F. P. Kooij, et al., Context-based path prediction for targets with switching dynamics, in IJCV, 2018, pp. 1-24.

[34] S. Pellegrini, et al., You'll never walk alone: Modeling social behavior for multi-target tracking, in ICCV, 2009, pp. 261-268.