Interaction Detection Between Vehicles and Vulnerable Road Users: A Deep Generative Approach with Attention

Hao Cheng, Member, IEEE, Li Feng, Hailong Liu, Member, IEEE, Takatsugu Hirayama, Member, IEEE, Hiroshi Murase, Fellow, IEEE, and Monika Sester

Abstract—Intersections where vehicles are permitted to turn and interact with vulnerable road users (VRUs) like pedestrians and cyclists are among some of the most challenging locations for automated and accurate recognition of road users’ behavior. In this paper, we propose a deep conditional generative model for interaction detection at such locations. It aims to automatically analyze massive video data about the continuity of road users’ behavior. This task is essential for many intelligent transportation systems such as traffic safety control and self-driving cars that depend on the understanding of road users’ locomotion. A Conditional Variational Auto-Encoder based model with Gaussian latent variables is trained to encode road users’ behavior and perform probabilistic and diverse predictions of interactions. The model takes as input the information of road users’ type, position and motion automatically extracted by a deep learning object detector and optical flow from videos, and generates frame-wise probabilities that represent the dynamics of interactions between a turning vehicle and any VRUs involved. The model’s efficacy was validated by testing on real-world datasets acquired from two different intersections. It achieved an F1-score above 0.96 at a right-turn intersection in Germany and 0.89 at a left-turn intersection in Japan, both with very busy traffic flows.

Index Terms—Interaction detection, vulnerable road user, deep generative model, sequence-to-sequence, deep learning

I. INTRODUCTION

In real-world traffic situations, it is not uncommon that heterogeneous road users like vehicles and vulnerable road users (VRUs, e.g., pedestrians and cyclists) have to directly interact with each other at some particular locations. Especially in city traffic, such type of locations include the turning areas of the so-called Turn-on-Red (TOR) intersections [1] or, more generally, intersections that allow vehicles to turn while other road users are crossing. During the time window of vehicles’ turning, their behavior is largely guided by social protocols, e.g., right-of-way or courtesy. For example, in Germany, as shown in Fig. 1 a turning vehicle at a permissive right-turn intersection often encounters cyclists that are passing by and pedestrians that are cross walking in the conflict areas. In Japan, a similar situation can be found at a permissive left-turn intersection [2] in left-hand traffic.

Efficiently and accurately learning how vehicles and VRUs interact with each other at such intersections is important for many applications. As statistics show, accidents often occur at places where vehicles and VRUs confront each other and there have been reports that VRUs were seriously injured by the overlook of car drivers at turning intersections [3], [4], [5]. Thus, one important application is the analysis of interactions and critical situations. In addition, the foreseeable advent of autonomous driving in urban areas [6], particularly in such locations, requires accurate recognition of road users’ behavior. Another potential application would be an accident warning system for road users.

Nowadays, with the ubiquity of traffic data and the development of computer vision techniques, there is a high chance for automatically recognizing road users’ behavior from massive video data. Hence, in this paper we aim to investigate an efficient way that can automatically analyze whether the continuity of road users’ behavior is interrupted when vehicles and VRUs meet at busy intersections, which we formalize as interaction detection by automatically extracting the user type, location and motion information from videos. The concept of interaction represents a changing level of reaction between road users. As defined by [7] an interaction is “a situation in which two or more road users are close enough in space and time and their distance is decreasing”.

Fig. 1: A right-turn intersection in Germany. A dedicated lane for cyclists is typically parallel to the crossing zone.

Hao Cheng, Li Feng and Monika Sester are with Leibniz University Hannover, Appelstrasse 9a, 30167 Hannover Germany. (e-mail: cheng@igk.uni-hannover.de; sester@igk.uni-hannover.de and li.feng1@outlook.com).

Hailong Liu and Hiroshi Murase are with Graduate School of Informatics, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8601, JAPAN. (e-mail: lhli81210@live.com and murase@is.nagoya-u.ac.jp).

Takatsugu Hirayama is with University of Human Environments, 6-2, Kamisanbonmatsu, Motojuku-cho, Okazaki, Aichi, 444-3505, JAPAN. (e-mail: t-hirayama@uhe.ac.jp).
Similarly, [8] describes an interaction between road users as “a continuum of safety related events”. Moreover, [9], [10] relate interaction to conflict that interaction can range from collision to negligible conflict risks. However, in everyday traffic, collisions or accidents fortunately only account for a very small amount—the tip of the pyramid of interactions [8], see Fig. 2. More frequent events are conflicts with different degrees of severity (serious, slight and potential) and undisturbed passages. Therefore, in this paper, a high level of classification is adopted for classifying events into non-interactions (undisturbed passages) and interactions (all the other events).

The task of interaction detection is to differentiate interaction and non-interaction levels over the dynamics of a vehicle turning sequence. Interaction is needed if the turning vehicle drives into an intersection while any VRUs approaching or moving in the intersection space (see Fig. 1). In order to avoid any conflicts that might happen at any time during the vehicle’s turning, they adapt their movement, i.e., velocity and orientation, accordingly. Otherwise, no interaction is needed if the target vehicle drives in an undisturbed manner with VRUs in its neighborhood, if there are any.

There are many challenges for automated interaction detection between vehicles and VRUs using video data. Road users’ behavior is dynamic and stochastic as they have to adjust their motion according to the reaction from each other. In addition, mixed types and varying numbers of roads users, as well as direct confrontations greatly complicate this task. The following open questions have to be addressed: (I) How to efficiently acquire, process, and label a large amount of video data for training a deep learning model for interaction detection considering all the relevant road users? (II) How can a system automatically detect the location and motion of the involved road users? (III) How to represent the dynamics of interactions in vehicle turning sequences of varying duration?

To tackle the above challenges, we propose a deep conditional generative model based on Conditional Variational Auto-Encoder (CVAE) [11] for automated interaction detection. The model is conditioned on the information extracted from video data and performs probabilistic inference for interaction prediction. As opposed to a discriminative model [12] that distinguishes interaction classes (interaction vs. non-interaction) from the observed information, a set of Gaussian latent variables are used to encode the dynamic and stochastic behavior patterns, which enables the generative model to perform diverse predictions at inference time. The contributions of this work are summarized as follows:

1) Various activities among all road user types were recorded using a camera at a right-turn intersection in Germany and a left-turn intersection in Japan for very busy traffic flows. They were processed for interaction detection in both right- and left-hand real-world traffic. In the future the data will be released for further research.

2) We combine a deep learning object detector to automatically detect all the relevant road users and optical flow to extract their motion. The combination captures the dynamics of all the road users and circumvents the tremendous work of manual tracking of trajectories.

3) Both sliding window and padding methods are explored to parse vehicle turning sequences of varying lengths.

4) We propose an end-to-end sequence-to-sequence conditional generative model with a self-attention mechanism [13] for interaction detection, which simultaneously takes both the object and motion information sequences and generates probabilities of interaction at each short interval (<0.1 s). The probabilities change accordingly when the intensity of interaction changes between a turning vehicle and the involved VRUs over time.

The remainder of the paper is organized as follows. Sect. II reviews the related studies on road user behavior at intersections. The proposed methodology is introduced in Section III. The detailed information of the datasets and evaluation metrics is provided in Sec. V. The experimental results are presented and analyzed in Sec. VI and the limitations of the model are further discussed in Sec. VII. Finally, the conclusions are drawn in Sec. VIII with potential directions of future work.

II. RELATED WORK

Early studies on road user behavior at intersections are focused on collision and conflict analyses [14], [15], [16], [17]. For example, [14] manually observed and studied a total of 25 collision scenes taped by videos at an intersection over a period of one year. [15] conducted a study of the safety impact of permitting turns on red lights (TOR) based on data of crashes. Examining actual collision and conflict scenes has many limitations. First, the occurrence of crash and accident events are very rare in daily traffic and vary from case to case, which cannot represent the majority of road users’ behavior [18] as undisturbed passages are not included. Second, as pointed out by [19], collision–based safety analysis is a reactive approach, which requires a significant number of collisions to be collected before an action is warranted. Third, those data are likely to be incompletely documented or protected for legal and privacy reasons, which leads to the data acquisition being complicated or even not possible. Most importantly, the above drawbacks make it almost impossible to automatically analyze the behavior of road users.

The development of computer vision techniques allows for automated analysis of road users’ behavior at intersections. The work by Ismail et al., [19], one of the early studies using computer vision techniques, automatically analyzed pedestrian–vehicle conflicts at an intersection using the extracted trajectories from video data. Later on, similarly, several works used trajectories extracted from videos...
to analyze before–and–after vehicle-pedestrian conflicts [20],
in street designs with elements of shared space [21], in less
organized traffic environments [22], and vehicle–bicycle
conflicts at intersections [18]. The work carried out by Ni
et al., [23] analyzed pedestrian–vehicle interaction patterns
using indicators of Time-to-collision (TTC) [24] and Gap
Time (GT) [19]. First, trajectories were extracted by the
semi-automated image processing tool Traffic Analyzer [25],
and then according to the TTC and GT values derived from the
trajectory speed profiles interactions were classified into three
classes: hard interaction, soft interaction and no interaction.
On the one hand, their work is very close to the studies carried
out in this paper. Both interactions and non-interactions are
studied at permissive right–turn intersections. On the other
hand, their work is not fully automated in terms of trajectory
extraction. Acquiring reliable trajectory data is often costly
and time-consuming. The quality of the data is difficult to
guarantee. For example, tracking multiple objects from frame
to frame is very challenging due to, e.g., abrupt object motion,
change of appearance and occlusions [26]. Errors and failures
in detection will propagate to the process of tracking, which
will later directly lead to wrong conclusions in the analysis step
[18]. Moreover, the above works only consider either
vehicle-pedestrian or vehicle-cyclist interactions. In real–world
traffic situations at big intersections other heterogeneous road
users are often involved at the same time.

In very recent years, deep learning methods have been
successfully applied to understand road users’ behavior at
intersections using video data, whereas many of them [27],
[28], [29], [30] are conducted from a perspective of a self-
driving car for pedestrian intent detection. With respect to
a third–person perspective, [12] trained an encoder-decoder
model to automatically detect interactions using sequences of
video frames from a static camera facing a very busy left–turn
intersection in a Japanese city. However, the discriminative
model is trained to optimize the reconstruction loss between
the pairs of ground truth and prediction, which tends to learn
the “average” behavior of road users. The dynamics and
stochastic behavior patterns are not fully captured. Hence, in
this paper, we propose a CVAE–based model with Gaussian
latent variables to encode various behavior patterns and per-
form diverse predictions. We not only test our model in left–
hand traffic but also right–hand traffic in different countries for
interaction detection between vehicles and all the other VRUs.

III. METHODOLOGY

This section explains the methodology of interaction detec-
tion in detail. Sec. III-A formulates the problem, Sec. III-B
describes the extraction of the input features, Sec. III-C
introduces the detection model, and Sec. III-D provides the
estimation of the model’s uncertainty.

A. Problem formulation

Interaction detection is formulated as a classification prob-
lem using the information extracted from videos. Given a
set of observed vehicle turning sequences of video frames
X = {X1, · · · , Xi, · · · }, the input of the i-th sequence is
characterized as X(T)i = {Xt|i|t=0,T−1}, where Xt ∈ IR(W × H × C)
is the frame at time step t, and T is the total number of observed
frames for the sequence. W, H and C denote the width,
height and the number of channels of each frame. Instead of
using raw images, object and optical–flow information (see
Sec. III-B for more details) is extracted from the frames and
is used as the sequence of the input. In this way, the per-
sonal information of road users,e.g.,face, gender, age, driving
plates can be protected. Yi is the corresponding ground truth
interaction label and Ỹi is the prediction.

Moreover, sequence-to-sequence modeling is applied to
learn the frame–wise dynamics of interactions over a turning
sequence. Similar to [28], the task defined above is a weakly
supervised learning problem due to the labelled data structure;
The interaction label is a dichotomous class that represents
the interaction level of the whole sequence. It does not provide
detailed information about how the interaction level changes with time. In fact, it is not feasible to manually label
each frame due to the tremendous amount of work. Without
knowing the exact fine-grained frame–wise interaction label,
the sequence–wise label is duplicated at each frame. Hence,
the form of the output is converted to obtain the time steps,
denoted as Y(T)i = {Y(T)i|t=0,T−1} for the i-th turning sequence.
Thereafter, the input and output are aligned at each frame for
a sequence-to-sequence modeling, denoted by Fig. 3 and Eq. (1).

\[
f(Y_i | X_i) = \lambda \sum_{t=0}^{T-1} f(Y_i^{(T)} | X_i^{(T)}),
\]

where f denotes the detection model and λ is a voting scheme
that summaries the frame–wise predictions to the sequence–
wise prediction. In this paper an average voting scheme that
weighs the prediction at each frame equally is adopted and
the sequence–wise prediction is the class label voted by the
majority [12]. The above conversion is based on the following
hypotheses: (1) Over a large dataset, sequence lengths vary
from one to another, which provides rich interaction informa-
tion of both long and short sequences. (2) The prediction error
between Yi and Ỹi is still computed at the sequence level
because each frame–wise prediction only partially contributes
to the sequence–wise prediction using the voting scheme.
This mechanism enables the model to automatically learn the frame-wise dynamics at each frame in training time.

In addition, sliding window and padding methods are proposed to deal with varying sequence lengths for training a recurrent neural network (RNN) based CVAE model, as shown in Fig. 3. This is because the most commonly used RNNs, e.g., Long Short-Term Memory (LSTM) [31], for sequence modeling often require a fixed sequence length. However, at an intersection some vehicles can quickly complete the turning if the space happens to be free, whereas some vehicles may have to wait for a long time to let VRUs cross first.

The sliding window method parses each sequence with a fixed window size \( w \). Eq. (2) denotes the sliding window method with a stride being the same as \( w \). The overlap between two consecutive windows is allowed when the stride is set to be smaller than the window size.

\[
X_i^{(T)} = \{ X_i^0, \ldots, X_i^{w1}, \ldots, X_i^{wk} \}, \quad \text{where} \quad k = \frac{T}{w}. \tag{2}
\]

The padding method uses zero-paddings at the end to extend the sequences shorter than a predefined length \( T^* > T \). The value of \( T^* \) can be adjusted to cover most of the sequences, e.g., \( T^* = \text{Max}\{T_1, \ldots, T_i, \ldots\} \), where \( T_i \) denotes the length of an arbitrary vehicle turning sequence. Meanwhile, a padding mask is used to annotate the exact sequence length so that the padded zero values are treated differently to mitigate the negative impact on the learning process.

\[
X_i^{(T)} = \{ X_i^0, \ldots, X_i^{T-1}, 0^T, \ldots, 0^{T^*-1}\}, \quad \text{Mask}^{(T)} = \{ 1^0, \ldots, 1^{T-1}, 0^T, \ldots, 0^{T^*-1}\}. \tag{3}
\]

After the formulation of the above sequence-to-sequence problem, we now introduce the pipeline of the proposed method for interaction detection, denoted by Fig. 4. It consists of two components: feature extraction (Sec. III-B) and the sequence-to-sequence CVAE model (Sec. III-C). Each component is explained in detail in the following subsections.

### B. Feature extraction

Object information and optical-flow information extracted from video frames are used as input features for the interaction detection task, as shown in Table I and Fig. 5.

| Feature | C1 | C2 | C3 | C4 | value |
|---------|----|----|----|----|-------|
| Object  | pedestrians | bikes/motors | cars/trucks | buses | \{0, 1\} |
| Optical flow* | orientation | 1 | velocity | \{0, 1\} |

*The HSV (Hue, Saturation, Value) color representation is used to store the optical-flow information. The hue channel (C1) is used to store orientation, the saturation channel (C2) is set to its maximum, and the value channel (C3) is used to store velocity. Note that there are four channels in each object frame, and only three channels in each optical-flow frame.

Object information contains road users’ type and location. The deep learning object detector, such as YOLOv3 [32] or M2Det [33], is leveraged for detecting all the relevant road users at each frame. Namely they are pedestrians, cyclists, motorbikes, cars, trucks, and buses. Different channels are used to store the road-user position information and each channel is dedicated to one or two similar road user types; Based on the acquired data that only very few motorbikes were detected, they are stored in the same channel for bicycles. Cars/trucks are stored in one channel given their very similar trace of turning, see Table I. The location of the detected road users (Fig. 5a) is mapped by the corresponding bounding boxes with values of one in each frame (Fig. 5c). Areas with no detected objects are set with values of zero, as shown in black color.

Optical flow is used to capture the motion of road users. It describes the distribution of velocities of moving pixels’
brightness in two consecutive images [34]. Moving objects are captured by optical flow and static objects and background are ignored. The dense optical–flow algorithm [35] is applied to map the displacement of moving objects and remove the static background information, see Fig. 5. Similarly, respective frame channels are dedicated to the information of orientation and velocity of the moving objects, see Table I.

The area of interest is the tuning space of the intersection and is marked by a binary mask (Fig. 4b), the other areas are not considered. As shown in Fig. 5a and 5b the mask of the area of interest slightly extends into the through lane next to the turning lane. Due to the oblique view of the camera, the upper body of the vehicles in the turning lane are partially projected into the through lane. The extended mask aims to include the upper body of the turning vehicles to be detected as well. The lower middle point of the bounding boxes of the detected vehicles is used to filter out the vehicles in the through lane. However, the extended mask introduces noise to the optical–flow information. For instance, as shown in Fig. 5d the motion of the vehicles in the through lane is also captured by the optical–flow, which cannot be easily filtered out given the irregular shapes and occlusion. But it later turns out that this noise is not problematic when both the object information and the optical–flow information are combined as the input for training the interaction detection model. Interactions between vehicles and VRUs only happen in the crossing zone.

C. CVAE model for interaction detection

The model of predicting the probabilities of interaction between a turning vehicle and the other crossing road users is denoted as \( f(Y|X) = \arg\max_{Y} p(Y|X, z) \), where \( f \) is a CVAE model that performs probabilistic prediction and \( z \) are the Gaussian latent variables. The model encodes the information of interaction into a latent space and predicts the interaction label \( \hat{Y} \) conditioned on the input \( X \) and \( z \). The variational lower bound \( \mathcal{L} \) of the model is given as follows:

\[
\log p_{\theta}(Y|X) \geq -D_{KL}(q_{\phi}(z|X, Y)||p_{0}(z)) + \mathbb{E}_{q_{\phi}(z|X, Y)}[\log p_{\theta}(Y|X, z)]. \tag{4}
\]

The model jointly trains a recognition model \( q_{\phi}(\cdot) \) (a.k.a. encoder), and a generative model \( p_{\theta}(\cdot) \) (a.k.a. decoder). In the training phase, the model is optimized via stochastic backpropagation [36]. \( q_{\phi}(\cdot) \) encodes the observed information and the ground truth label into the latent variables \( z \). In other words, the inserted label in training is combined with the condition to parameterize the Gaussian latent space, which later can be used for structured prediction to map the many possible outputs \( \hat{Y} \). \( p_{\theta}(\cdot) \) decodes the prediction of the interaction label conditioned on the input and the latent variables. \( -D_{KL}(\cdot) \) is the negative Kullback-Leibler divergence of the approximate posterior from the prior \( p_{0}(z) \) and acts as a regularizer, which pushes the approximate posterior \( q_{\phi}(\cdot) \) to the prior distribution \( p_{0}(z) \). Note that in our model the prior is relaxed to make the latent variables statistically independent from the input variables so that \( p_{\theta}(z) = p_{\theta}(z|X) \) [37]. The prediction loss \( \mathbb{E}_{q_{\phi}(z|X, Y)}[\cdot] \) measures the distance between \( Y \) and \( \hat{Y} \). The binary cross–entropy loss is used as the prediction loss, as denoted by Eq. (5).

\[
\mathcal{H}(\hat{Y} - Y) = -(Y \log \hat{Y} + (1 - Y) \log(1 - \hat{Y})). \tag{5}
\]

In the inference phase, the decoder predicts the interaction label conditioned on the input of the observed information concatenated with a latent variable directly sampled from the Gaussian prior \( p_{0}(z) \). The sampling process is done multiple times to perform diverse predictions [11].

Convolutional Neural Networks (CNNs) and RNNs, as well as the self-attention mechanism [13] are employed in the CVAE model to learn the parameters of \( \theta \) and \( \phi \). As shown in Fig. 4 the encoder has two branches: X-Encoder and Y-Encoder. They are dedicated to extracting low–level features from the condition (the object and optical–flow information) and the interaction label information, respectively. Each module, i.e., X-Encoder, Y-Encoder, Latent Space, and Decoder of the CVAE model, is explained in detail as follows.

The X-Encoder manipulates two CNNs for learning spatial features from the object frame sequence and the optical–flow frame sequence, respectively. Without loss of generality, the object frame sequence using the sliding window (e.g., \( w = 8 \)) method is taken as an example for explaining the learning process. First, each frame from the sliding window is passed to a CNN to learn spatial features. As shown in Fig. 3 the CNN has three 2D convolutional (CONV) layers with each one followed by a Maximum Pooling (MP) layer and a Batch Normalization (BN) [38]. It takes the frame that contains object information as input and outputs a flattened feature vector. This process is done frame by frame for all the frames in the sliding window. Then, the output feature vectors of all the frames are timely distributed as a sequence that maintains the same length as the window size, as shown in Fig. 4 for the X-Encoder. The optical–flow frame sequence is processed by

\[1\] This process works in the same way for the padding method with the predefined sequence length, instead of the sliding window size.
class at each frame. The output of the Decoder are the frame–wise predictions of the interaction class. In the end, the average voting scheme is used to summarize the frame–wise predictions to get the sequence–wise prediction for the interaction class.

In inference time, the interactions for unseen vehicle turning sequences are classified using the trained CVAE model. First, the object and optical–flow information are encoded by the X-Encoder. A latent variable is sampled from the Gaussian distribution. Then the Decoder generates the probabilities of the interaction class for each sequence conditioned on the output of the X-Encoder and the sampled latent variable. The sampling is repeated multiple times at each step so that the Decoder generates diverse probabilities of the interaction class.

D. Estimation of uncertainty

Kernel density estimation (KDE) \cite{39, 40} is used to measure the uncertainty of the diverse predictions generated by the above multi-sampling process. At each frame, the predictions \( \{Y_{t,1}, \ldots, Y_{t,N}\} \) are assumed to be i.i.d. samples drawn from an unknown density function \( g(Y) \). \( N \) is the total number of predictions and \( t \leq T \), and \( T \) is the total steps of the give sequence \( i \). The KDE is calculated as:

\[
\hat{g}(Y)^t = \frac{1}{N} \sum_{i=1}^{N} K_h(Y - Y_i) = \frac{1}{Nh} \sum_{i=1}^{N} \frac{K(Y - Y_i)}{h},
\]

where \( K(\cdot) \) is the Gaussian kernel function, \( h \) is the smoothing parameter (also called bandwidth). The log-likelihood of the average prediction at step \( t \) is determined by \( \mathcal{L}(\log(\hat{g}), \bar{Y}_i)^t \), where \( \bar{Y}_i \) is the average prediction. The uncertainty is defined as the residual of the normalized log-likelihood averaged over all the steps for the given sequence \( i \), as denoted by Eq. (7):

\[
\Gamma_i = 1 - \frac{1}{T} \sum_{t=0}^{T-1} \omega \mathcal{L}(\log(\hat{g}), \bar{Y}_i)^f,
\]

where \( \omega \) is the normalization parameter that scales the values to \([0,1]\) and \( \Gamma_i \) stands for the degree of uncertainty for the prediction over the sequence \( i \).

IV. DATA ACQUISITION AND PRE-PROCESSING

Real–world datasets were acquired to test the performance of the proposed model for interaction detection. Fig. 4 shows the screenshots of the two intersections where various traffic scenes were recorded. The KoW dataset was acquired by \cite{41} from a very busy right–turn intersection in a German city. The videos recorded traffic conditions from 00:02 a.m. to 11:58 p.m. on November 8th and 9th, 2019 in Hannover. The videos were recorded in 1280 × 720 pixels at 25 fps by a camera module (Raspberry Pi Camera Module v2) installed inside a building (ca. 20 m ground elevation) facing the intersection and stored in .h264 format. We use an approximately 14-hour sub-footage from two seven–hour segments (8 a.m. to 3 p.m. on both the 8th and 9th), when there was enough traffic and adequate ambient light to perform stable image processing for feature extraction. The NGY dataset was provided by Nagoya Toyopet Corporation. It was acquired from an extremely busy
left–turn intersection in a Japanese city. In total, approximately 24 hours of traffic footage from an oblique view at one of the major intersections in Nagoya were recorded from 11 a.m. to 11 a.m. on April 23rd and 24th, 2019. The videos were recorded in 1600 × 1200 pixels at 30 fps using a camera (Panasonic WV-SF781L) installed inside a building (ca. 3 m ground elevation) adjacent to the intersection and stored in .mp4 format. Similarly, we use a twelve-hour sub-footage recorded from 11 a.m. to 6 p.m. on the 23th and from 6 a.m. to 11 a.m. on the 24th.

Both datasets were pre-processed for later usage. Due to the missing camera intrinsic and extrinsic parameters, no projection was done for extracting the trajectory data. The pre-process aimed to identify vehicle turning sequences and extract all the road users’ type, position and motion information. First, two annotators for each dataset manually detected the sequence scenes where a vehicle turned right at the KoW intersection or left at the NGY intersection, and extracted the time intervals of the vehicle staying in the yellow contours (see Fig. 7). The annotators independently determined whether or not interactions occurred in each scene, afterwards they revised and agreed with each other and labeled each scene as “non-interaction” or “interaction”. Then, YOLOv3 [32] and M2Det [33] were used to detect all the traffic related objects at the original frame rate of the KoW (25 fps) and NGY (30 fps) datasets, respectively. Note that these two sources of data were from different providers so that the camera settings and the object detection algorithms were not unified. Considering the change between two consecutive frames is small, a possibly failed detection in the current frame is supplemented by the detection in the previous or the next frame if either of them was available. Otherwise, the sequences with failed detection and no supplementation were discarded. In addition, the dense optical–flow algorithm [35] was used to extract the optical–flow information from the sequences. Different from the object detection, the frame rate was down–sampled to the half of the original frame rate, i.e., 12.5 fps for KoW and 15 fps for NGY. This aims to reduce the computational cost and increase the offset of moving pixels between two consecutive frames, so as to improve the extraction performance [35] of optical flow. In the end, both the object and the optical–flow sequences were aligned with the down–sampled frame rate in each dataset, which is used as the time step for interaction detection.

The data processing yields over 2000 vehicle turning sequences with varying lengths, as denoted by Fig. 8. Within each dataset, sequence lengths measured in seconds are very different. The non-interaction sequences are significantly shorter than the interaction ones (U-test, $U = 382142, p \ll 0.01$ for KoW and $U = 67199, p \ll 0.01$ for NGY), and the standard deviation in each class in both datasets deviates in a large range. This indicates that the duration of a sequence is not an accurate feature for the detection task; a short sequence duration does not necessarily imply no interaction. In addition, the sequence lengths of all the sequences over each dataset are very unevenly distributed, i.e., long–tail distributed, especially for the NGY dataset, see Fig. 8b. Across the datasets, the sequences in the KoW and NGY datasets are different in terms of not only travel direction but also frame size and rate, as well as sequence length in general. Though non-interaction sequences from both datasets have similar sequence lengths, i.e., on average non-interaction sequences have a length of $2 \text{s}$ in KoW and $3 \text{s}$ in NGY, interaction sequences in NGY have a longer average sequence length ($10.8 \text{s}$) than the ones in KoW ($7.2 \text{s}$). Due to the higher density of traffic at the NGY intersection compared to the KoW intersection, vehicles often had to wait for more pedestrians and cyclists crossing. The above differences make cross dataset validation very difficult (more details in Sec. VII-B).

The acquired datasets were further prepared for training the detection models, which involves sample balancing, sequence padding and dataset partitioning. The number of samples in each class was balanced to perform unbiased training. For both datasets, the maximum number of sequences in each class was

---

Footnote:

²Less than 1 % of the sequences were initially annotated differently.
set to a value that is close to the capacity of the smaller class. Note that the small amount of very long sequences (i.e., > 500 frames, see Fig. 8b) were not used for the experiments. All such sequences are from the class of interaction, i.e., vehicles had to wait for a long time to let other road users cross the intersection. The removal of these sequences balances sample size and length in both classes, in order to prevent a model being biased towards the interaction class. Sequences with a smaller number of frames than the threshold $T^*$ are padded with zeros for the padding method (see Sec. [I]I-A). However, if $T^*$ is too large most of the sequences will be padded with zeros and this will lead to noisy samples; if $T^*$ is too small many long sequences will be excluded. To balance the trade-off, based on the sequence length distributions over the datasets (Fig. 8b), $T^*$ was set to 100 frames for both datasets so that the majority of all the sequences were included. The sequences shorter than or equal to $T^*$ were preserved for the experiments of the models that use the padding method. Sequences longer than $T^*$ exceed the maximum length of the input size that the models can handle. Therefore, they were discarded. Under the balanced criteria above for each class, both the datasets were then randomly split into training and test sets by the ratio of \(70 : 30\). Additionally 20\% of the training data was separated as an independent validation set to monitor the process of training. Table IV lists the statistics of the final data used for the experiments after the preparation steps.

V. Experiments

A. Baseline and ablative models

To evaluate the performance of the proposed CVAE model, it is compared with a baseline model. The baseline model is a sequence-to-sequence encoder–decoder model that uses the same input features from the object and motion information for interaction detection [12]. It has the same structure as the X-Encoder and the Decoder that are implemented in the CVAE model (Fig. 4). The difference between these two models is the sample generation process. The baseline model is a discriminative model and does not use the class label information and the conditional information for learning the latent variables that mimic the stochastic behavior in vehicle–VRU interactions. Without randomly sampling from the Gaussian latent variables, the output of the sequence-to-sequence encoder–decoder model is deterministic.

A series of ablative models are designed to analyze the contribution of the object information ($ob$), the optical–flow information ($op$), and the self-attention mechanism ($att$). The ablative models are trained by removing one of the aforementioned parts, as denoted in Table III.

---

| Name | Input form | Max. length* | Training | Validation | Test | Total |
|------|------------|--------------|----------|------------|------|-------|
| KoW sliding | 500 | 360/360 | 90/90 | 192/192 | 642/642 |
| KoW padding | 100 | 352/352 | 88/88 | 188/188 | 628/628 |
| NGY sliding | 500 | 291/291 | 74/74 | 159/159 | 530/530 |
| NGY padding | 100 | 132/132 | 33/33 | 70/70 | 235/235 |

| Model name | ($ob$) | ($op$) | ($att$) | Sample generation |
|-------------|-------|-------|-------|-------------------|
| [$S+ob+op+att$] | ✓ | ✓ | ✓ | ✓ |
| [$C+op+att$] | ✓ | ✓ | ✓ | ✓ |
| [$C+ob+att$] | ✓ | ✓ | ✓ | ✓ |
| [$C+ob+op$] | ✓ | ✓ | ✓ | ✓ |
| [$S+ob+op+att$] | ✓ | ✓ | ✓ | ✓ |

1 the baseline model; 2 the complete CVAE model

B. Evaluation metrics

Tested samples are categorized according to the comparison between their ground truth and the predicted labels, as listed in Table IV. Accuracy, Precision, Recall and F1-score are applied to measure the performance of interaction detection on the test data from both the KoW and NGY intersections.

| Category name | Ground truth | Prediction |
|---------------|--------------|------------|
| TP: true positive | interaction | interaction |
| TN: true negative | non-interaction | non-interaction |
| FP: false positive | non-interaction | interaction |
| FN: false negative | interaction | non-interaction |

Accuracy is the fraction of the number of the correctly predicted samples over the total number of samples.

\[
\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.
\]

Precision is the fraction of the number of the TP samples over the number of predicted positive samples.

\[
\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.
\]

Recall is the fraction of the number of the TP samples over the number of actual positive samples in the whole dataset.

\[
\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.
\]

F1-score is used to provide a measurement of the overall performance of a model. It is defined as the so-called harmonic mean of precision and recall.

\[
\text{F1-score} = 2 \times (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall}).
\]

C. Experimental Settings

The kernel size of the CNNs in each layer is set to 8, 4, and 2 respectively with a stride of 2 and the same padding for the borders. The size of the first hidden layer of the LSTM is set to 64 and the second hidden layer is 32. The size of the latent variables is set to 64. All the models are trained by a learning rate of \(10^{-4}\) with a zero decay using the Adam optimizer (\(\beta_1 = 0.9\) and \(\beta_2 = 0.999\)) [42]. The batch size is set to 32, and all the models were trained for 50 epochs on an NVIDIA Quadro T2000 GPU. In inference time, the number of sampling \(N\) is set to 100 for all the CVAE-based models.

VI. Results

This section presents the quantitative and qualitative results for each intersection, as well as the discussion of the results.
A. Quantitative results

The quantitative results are summarized in Table V and VI for the right–turn KoW intersection and the left–turn NGY intersection, respectively. Due to the multi–sampling process the results of the CVAE–based models are not deterministic, hence the corresponding standard deviations are provided.

Table V shows the results of the interaction detection at the right–turn intersection. (1) Both the sliding window and padding methods yield similar and very accurate results for interaction detection using the combined information from object detection and optical flow. The accuracy and F1-score are both above 0.95. (2) Compared to the baseline models [S2S+ob+op+att], the proposed models [CVAE+ob+op+att] have a slightly better performance using the sliding window method and comparable performance using the padding method. (3) Compared to the ablative models, the combined information improves the performance using the sliding window method. However, the improvement is not obvious using the padding method, especially compared to the ablative model that only uses the object information. On the other hand, regardless of the sliding window method or padding method, the ablative models that merely use optical–flow information only achieve an accuracy below 0.70. (4) The self-attention mechanism does not lead to an obviously better or worse performance using either the sliding window or padding method.

TABLE V: Detection results of the right–turn intersection on the KoW dataset. Best values are highlighted in boldface.

| Model       | shape | Accuracy | Precision | Recall | F1-score |
|-------------|-------|----------|-----------|--------|----------|
| [S+ob+op+att] | sli.  | 0.951    | 0.935     | 0.969  | 0.951    |
| [C+op+att]   | sli.  | 0.692±0.006 | 0.717±0.007 | 0.635±0.011 | 0.673±0.008 |
| [C+ob+att]   | sli.  | 0.952±0.002 | 0.934±0.002 | 0.973±0.004 | 0.953±0.002 |
| [C+ob+op]    | sli.  | 0.965±0.001 | 0.976±0.003 | 0.953±0.00 | 0.964±0.001 |
| [C+ob+op+att] | sli.  | 0.961±0.002 | 0.969±0.004 | 0.953±0.00 | 0.966±0.002 |
| [S+ob+op+att] | pad.  | 0.963    | 0.944     | 0.984  | 0.964    |
| [C+op+att]   | pad.  | 0.610±0.008 | 0.649±0.011 | 0.479±0.012 | 0.551±0.010 |
| [C+ob+att]   | pad.  | 0.967±0.002 | 0.955±0.003 | 0.980±0.003 | 0.967±0.002 |
| [C+ob+op]    | pad.  | 0.966±0.003 | 0.946±0.004 | 0.980±0.001 | 0.967±0.002 |
| [C+ob+op+att] | pad.  | 0.962±0.002 | 0.955±0.003 | 0.973±0.00 | 0.963±0.002 |

Table VI shows the results of the interaction detection at the left–turn intersection. (1) Both the sliding window and padding methods yield reasonable results for interaction detection using the combined information. However, the predictions of the sliding window method are more accurate than the padding method. (2) Compared to the baseline models, the CVAE models using the combined information achieve better performance, especially by using the sliding window method (e.g., about 0.05 increment in F1-score). (3) Compared to the ablative models, the improvement by using the combined information can be found in both the sliding window and padding methods. (4) The best performance, especially measured by recall (0.916) and F1-score (0.892) on the NGY dataset, is achieved by the proposed CVAE model using the sliding window method with the self-attention mechanism.

The Kernel Density Estimation (KDE) function (see Sec. II-D) is used to measure the uncertainty levels of the CVAE–based models with different input structures. The uncertainties of the CVAE–based models are plotted in Fig. 9.

TABLE VI: Detection results of the left–turn intersection on the NGY dataset. Best values are highlighted in boldface.

| Model                     | Shape | Accuracy  | Precision | Recall | F1-score |
|---------------------------|-------|-----------|-----------|--------|----------|
| [S+ob+op+att]             | sli.  | 0.849     | 0.878     | 0.811  | 0.843    |
| [C+op+att]                | sli.  | 0.878±0.004 | 0.854±0.004 | 0.912±0.006 | 0.882±0.004 |
| [C+ob+att]                | sli.  | 0.734±0.008 | 0.695±0.007 | 0.824±0.009 | 0.759±0.007 |
| [C+ob+op]                 | sli.  | 0.882±0.006 | 0.915±0.004 | 0.842±0.009 | 0.887±0.006 |
| [C+ob+op+att]             | sli.  | 0.889±0.005 | 0.860±0.005 | 0.916±0.007 | 0.892±0.004 |
| [C+ob+op+att] pad.        | 0.721 | 0.712     | 0.743     | 0.727  |
| [C+op+att] pad.           | 0.764±0.010 | 0.756±0.001 | 0.808±0.010 | 0.781±0.009 |
| [C+ob+att] pad.           | 0.683±0.012 | 0.661±0.011 | 0.751±0.010 | 0.703±0.013 |
| [C+ob+op] pad.            | 0.782±0.007 | 0.763±0.010 | 0.819±0.014 | 0.790±0.009 |
| [C+ob+op+att] pad.        | 0.742±0.007 | 0.750±0.009 | 0.728±0.010 | 0.739±0.007 |

Fig. 9: Uncertainty measure of the CVAE–based models tested on the KoW and NGY datasets. The mean value is denoted by the yellow square in each box-plot. The uncertainty levels on the KoW and NGY datasets. The mean value is denoted by the yellow square in each box-plot. The uncertainty levels across the models are compared using the Mann-Whitney U-test. p-values are annotated using * or ns (not significant), where ns: 0.05 < p < 1.00, *: $10^{-2}$ < p < 0.05, **: $10^{-3}$ < p < $10^{-2}$, ***: $10^{-4}$ < p < $10^{-3}$, and ****: p <= $10^{-4}$.

and compared by the Mann-Whitney U-test. Fig. 9a and 9b demonstrate that the CVAE–based models [CVAE+op+att] using only the optical–flow information generate significantly more uncertain predictions than the other models tested on the KoW dataset. This pattern is consistent with the prediction performance that they also yield less accurate predictions. A similar pattern can be observed from the CVAE–based models [CVAE+ob+att] using only the object information (Fig. 9c and 9d) tested on the NGY dataset. When the uncertainty level is high in the predictions, the accuracy level also drops.

The confusion matrices for the proposed CVAE model using both the object and the optical–flow information are presented in Fig. 10. It can be seen that the model using either the sliding window (true negative rate 0.970 and true positive rate 0.953) or the padding method (true negative rate 0.951 and true positive rate 0.973) achieves high performance for interaction detection tested on the KoW dataset. The proposed model...
achieves good performance using the sliding window method (true negative rate 0.861 and true positive rate 0.916) on the NGY dataset and maintains a relatively low false negative rate (0.084). Nevertheless, the performance of the proposed CVAE model tested on the NGY dataset is inferior to the one on the KoW dataset. In contrast, the padding method only achieves mediocre performance (true negative rate 0.757 and true positive rate 0.728) tested on the NGY dataset.

**B. Qualitative results**

The qualitative results intuitively showcase the process of interaction detection of the models. The fine-grained probability of the predicted interaction at each frame provides a clue of how the interaction intensity evolves over time.

Fig. 11 demonstrates a non-interaction scenario at the KoW intersection between the right-turning target vehicle (in the blue bounding box) and the standstill pedestrian (in the red bounding box). There was no explicit interaction between them as the continuity of their behavior was not affected when the gap between them closed up, so the sequence was annotated as no interaction required. The sequence-level prediction is the average vote of all the frame-level predictions. At the sequence level, all the models correctly predict this scenario as non-interaction using both the sliding window (Fig. 11a) and padding (Fig. 11b) methods, except the ablative model [CVAE+ob+op+att] (in black) that only uses the optical-flow information. However, all the models predict a high probability of interaction when the vehicle approached the pedestrian. Also, the variance of the CVAE–based models in Fig. 11b increases when the probabilities change from under 0.5 to a higher value of interaction. The baseline model

[SGS+ob+op+att] (in black) generates a similar pattern in frame-wise predictions. But it is deterministic at each frame and does not have the mechanism to represent the uncertainty of the predictions.

Fig. 12 demonstrates an interaction scenario at the NGY intersection between the left-turning target vehicle (in the blue bounding box) and the crossing cyclist (in the red bounding box). Interaction was required between them as the vehicle had to decelerate or even briefly stop, yielding the way to the cyclist. All the models correctly predict this sequence as interaction using both the sliding window (Fig. 12a) and padding (Fig. 12b) methods. Similar to the scenario above, the variance of the probabilities for the CVAE–based models using the sliding window method change with the modification of the distance between the target vehicle and the cyclist. As the distance between them decreases, the probability is higher and the variance is smaller for interaction, and vice versa.
the number of training samples was not compromised for the experiments. On the other hand, the padding method requires a pre-defined fixed sequence length, which is unable to deal with longer sequences. Hence, the number of training samples was compromised by excluding longer sequences. The impact of the training data size has been shown by the performance difference across the KoW and NGY datasets. The numbers of the training samples of KoW for the sliding window and padding methods are similar (see Table II), and their performances for interaction detection are comparable to each other (see Table V). On the contrary, the number of training samples of NGY for the sliding window method is much larger than the one for the padding method (see Table II). The prediction by the sliding window method is more accurate than the padding method (see Table VI). In addition, the shorter sequences were padded with zeros. This is problematic for the information extracted by optical flow. Because the zero values in the optical–flow feature vector represent the background of the intersection or static road users. Even though a padding mask is incorporated into the sequence for indicating the actual sequence length, the negative impact cannot be fully remedied due to the complex learning process in training. The negative impact of padded zeros from the padding method has been uncovered by the impaired performance of the ablative model \( \text{CVAE+op+att} \) compared to the sliding window method.

In general, the proposed CVAE model \( \text{CVAE+ob+op+att} \) outperforms the baseline model \( \text{S2S+ob+op+att} \) quantitatively (see Table V) and qualitatively (see Fig. 11 and 12). In the CVAE model, the latent variables \( z \) are trained to capture the stochastic attributes of road users’ behavior in various traffic situations, which is optimized by the Kullback-Leibler divergence loss against a Gaussian prior. In addition to the Kullback-Leibler divergence loss, the reconstruction loss is trained by minimizing the cross-entropy loss between ground truth and prediction. Optimizing these two losses together enables the CVAE model to generate diverse predictions. With the multi–sampling process of the latent variables, the predicted probabilities of interaction at each frame vary, especially when the probabilities change over time, see Fig. 11 and 12. The variance of the probabilities indicates the uncertainty in the predictions. In contrast, the baseline model is trained only by optimizing the reconstruction loss. It tends to learn the “average” behavior of road users. Predictions by the baseline model are rather deterministic and there is no mechanism to interpret the uncertainty of the predictions.

The combined information of object detection and optical flow shows a stable performance for the interaction detection task. The performance of interaction detection highly depends on the quality of the input information extracted from videos, which is often impaired for many reasons. A single type of information may not be sufficient for this task. As indicated by the limited performance of the ablative models that only use optical–flow information on the KoW dataset, without the object information the noisy optical–flow information from the through lane or padded zeros may impact the detection performance. Similarly, the ablative models that only use object information on the NGY dataset also achieved limited performance. The distorted object information, especially for
the road users close to the camera at the NGY dataset, could lead to wrong interaction detection. The combination of the extraction techniques increases the possibility to maintain a good quality of the input information, so as to achieve a stable performance of interaction detection.

The self-attention mechanism does not show a consistent benefit across the datasets. The CVAE models regardless of the self-attention mechanism yield very similar results for interaction detection using both the sliding window and padding methods on the KoW dataset, and using the padding method on the NGY dataset. The improvement with the self-attention mechanism can be found for the sliding window method on the NGY dataset. First, the self-attention layer is followed by an LSTM (Fig. 4), which may be redundant for learning the interconnections along the time axis. The self-attention layer is likely under-trained due to the small dataset size or redundant layers, whereas the LSTM is already sufficient for learning the temporal patterns of the sequence data from the KoW intersection. On the other hand, the sequence data from the NGY intersection is more complex, e.g., longer and more varying sequence lengths (see Fig. 8) and high traffic density. On top of the LSTM, the self-attention mechanism has turned out to be beneficial for further learning the temporal patterns.

In summary, the sliding window method is more flexible than the padding method in dealing with various sequence lengths. The CVAE models using the combined information of both object detection and optical flow achieve a more stable performance compared to using a single type of information. The multi-sampling process enables the CVAE-based models to mimic the uncertainty of road users’ behavior, and the self-attention mechanism is only beneficial for learning temporal patterns from complex data. Overall the proposed model [CVAE+ob+op+att] using the sliding window method achieves a more desirable performance across the datasets.

VII. DISCUSSION

Here we discuss the failed detection by the proposed CVAE model using the sliding window method and the challenges to transfer the model from one intersection to the other smoothly.

A. Failed detection

Various reasons can lead to a wrong interaction classification. Table VII categorizes the wrongly detected scenarios i.e., false negative (FN) and false positive (FP) tested on both the KoW and NGY datasets. The false negative examples are visualized in Fig. 13 for KoW and in Fig. 14 for NGY, and the false positive examples are visualized in Fig. 15 for KoW and in Fig. 16 for NGY.

The FN scenarios are associated with VRUs entering (FN-I and FN-III) or leaving (FN-II) the intersection space. Due to their relatively long distance to the target vehicle, but fast travel speed, they are erroneously classified as non-interaction.

Most of the FP scenarios are associated with the target vehicle following a leading vehicle. As exemplified by FP-I in Fig. 15 and 16 only the leading vehicle (in the yellow bounding box) required direct interactions with the involved VRUs. After the leading vehicle finished turning, the pedestrian (in the red bounding box) also completed crossing. Afterwards, there was no interaction required from the target vehicle (in the blue bounding box) with the VRUs. However, the CVAE model has limited performance in handling this type of situations. Because the current model does not have explicit information to differentiate the leading and target vehicles, and the model is not specifically trained for car following situations.

In addition, a short distance from the VRUs to the intersection (FP-II Fig. 16) or just finishing crossing (FP-IV, Fig. 15 and 16), can also lead to an FP case. The distortion of the distance may lead to an FP case as well. For example, in FP-III in Fig. 16 even though the pedestrian on the sidewalk was relatively far from the turning vehicle, it has been still classified as an interaction by the model due to the distorted

| Errors | Scenario description | Category | KoW | NGY |
|--------|----------------------|----------|-----|-----|
| FN     | pedestrian entering the intersection (FN-I) | 8        | 7   |
| FN     | pedestrian leaving the intersection (FN-II) | 1        | 4   |
| FN     | cyclist entering the intersection (FN-III) | -        | 2   |
| FP     | car following (FP-I) | 4        | 17  |
| FP     | pedestrian standing near the intersection (FP-II) | -        | 3   |
| FP     | pedestrian approaching from the sidewalk (FP-III) | -        | 1   |
| FP     | pedestrian finishing crossing (FP-IV) | 1        | 1   |

Total* 14 35

*The total number of the wrongly detected scenarios listed here is slightly different as shown in the above confusion matrices due to the multi-sampling of the CVAE model.

Fig. 13: Examples of the false negative detection on the KoW dataset. The right–turning target vehicles are denoted by the blue bounding boxes and the involved VRUs are denoted by the red bounding boxes.

Fig. 14: Examples of the false negative detection on the NGY dataset. The left–turning target vehicles are denoted by the blue bounding boxes and the involved VRUs are denoted by the red bounding boxes.
distance at the NGY intersection. However, the camera at the KoW intersection was installed at a higher elevation than the camera at the NGY intersection. The distortion is thus less harmful for the horizontal distance. Among other reasons, this might have contributed to the better performance of the model tested on the KoW dataset than on the NGY dataset.

Based on the discussion of the failed detection scenarios, the limitations of the proposed model are summarized as follows: 1) The definition of interaction only considers the relationship between the target turning vehicle and the involved VRUs. The car-following relationship is not included. Without considering this relationship often leads to false positive detection between the following car and the VRUs. 2) The crossing directions of VRUs are not used as a factor to differentiate interaction types. For example, interactions between a turning vehicle and pedestrians or cyclists that are approaching the crossing area from near side and far side are labeled as the same interaction type. However, the discussion above indicates that the moving directions of VRUs are important for estimating the interactions between the turning vehicle and VRUs, especially when the VRUs are leaving the intersection. 3) The exact distances between a turning vehicle and the involved VRUs are not measured, thus the change of the distances between them cannot be correctly quantified. Without the measurement of distance, it is difficult for a model to distinguish the subtle difference between interaction and non-interaction. Especially, when the image distance is distorted by the camera’s perspective or when an occlusion happens, the model’s performance will be impaired.

B. Challenges of cross dataset generalization

The models proposed in this chapter are adapted for interaction detection at different intersections, in order to analyze the generalizability of the above models. In the previous experiment setting, all the models were trained and tested using the dataset from the same intersections. In this section, the models trained using the KoW dataset were tested on the NGY dataset, and vice versa. Frames from the test set were resized into the same size and mirrored into the same direction as the training set so that the trained models could be tested on both datasets without changing the setting of the input size.

Table VIII lists the results for the cross dataset validation. It can be seen that, both the CVAE-based and sequence-to-sequence encoder–decoder models do not achieve good performance either using the sliding window or padding method. This could be because these two datasets (Sec. IV) are very different in terms of, e.g., vehicle’s travel direction, camera’s perspective, frame size and rate, sequence length, intersection layout, traffic density, and cultural factors (Germany vs. Japan). Note that, because the camera parameters and reference coordinates were not available from the datasets, in this paper projection is not applied to transform the perspective to a bird’s-eye view. Under the cross dataset validation setting, the resized frames distort the motion and position of the dynamic objects and confused the models for predicting the interactions between vehicles and VRUs. However, this leads to the future research question—how to generalize the models for different intersections and traffic, and even different cultures?

TABLE VIII: Performance of cross dataset validation for interaction detection on the KoW and NGY datasets.

| Model Shape | Accuracy | Precision | Recall | F1-score |
|-------------|----------|-----------|--------|----------|
| [S+ob+op+att] sli. | 0.490 | 0.430 | 0.490 | 0.350 |
| [S+ob+op+att] pad. | 0.473 | 0.450 | 0.470 | 0.400 |
| [C+ob+op+att] sli. | 0.490±0.002 | 0.491±0.001 | 0.804±0.004 | 0.609±0.003 |
| [C+ob+op+att] pad. | 0.490±0.002 | 0.475±0.002 | 0.174±0.001 | 0.255±0.003 |

VIII. CONCLUSION

In this paper, an end-to-end sequence-to-sequence generative model based on CVAE has been proposed to automatically detect interactions between vehicles and VRUs at intersections using video data. All the road users that appear during a vehicle’s turning time are detected by a deep learning object detector, and their motion information is captured by optical flow, simultaneously. The sequences of object detection and optical-flow information together provide rich information for interaction detection. Both sliding window and padding methods are explored to learn dynamic patterns from turning sequences of varying lengths. The proposed model predicts
fine-grained interaction class label at each frame of less than 0.1 s, which provides a clue of how the intensity of an interaction between a turning vehicle and VRUs evolves as time unfolds. The average voting scheme summarizes the frame-wise predictions so as to accurately get a class label for the overall sequence. Besides, the multi-sampling process generates diverse predictions and the Kernel Density Estimation function is used to measure the uncertainty level. The efficacy of the model was validated at a right-turn intersection in Germany and a left-turn intersection in Japan. It achieved an F1-score above 0.96 at the right-turn intersection and 0.89 at the left-turn intersection, and outperformed a sequence-to-sequence encoder-decoder model quantitatively and qualitatively.

Furthermore, a series of ablation studies investigated the effectiveness of the combined information from object detection and optical flow, and the self-attention mechanism for learning temporal patterns from complex sequences. The comparison between the sliding window and padding methods showed that the former method is more flexible in coping with sequences of varying sequence length—the number of samples is not restricted to the maximum sequence length that a model can handle, which stands in contrast to the padding method. The self-attention mechanism has only shown a clear positive effect for interaction detection on the complex NGY dataset.

In future work, several improvements can be made to reduce the limitations of the detection model. First, the dichotomous classification of interaction should be extended to multi-class classification, e.g., taking the confrontation direction and car-following relationship into consideration. Second, the accuracy of feature extraction can be enhanced by using multiple cameras or even tracking. Third, projective transformation techniques or data recorded by drones with a bird’s-eye view can be explored to reduce the distortion caused by the camera’s perspective and filter the noisy optical-flow information captured from the through lane next to the turning lane. Last but not least, the generalizability of the model for interaction detection at different intersections needs to be further studied.

ACKNOWLEDGMENTS

The project is funded by the German Research Foundation (DFG) through the Research Training Group SocialCars (227198829/GRK1931). This work is a collaboration with Murase Lab at Nagoya University and supported by Nagoya Toyopet Corporation with the Nagoya intersection dataset.

REFERENCES

[1] H. McGee and D. Warren, “Right turn on red,” Public Roads, vol. 40, no. HS-019 049, 1976.
[2] W. K. Alhajyaseen, M. Asano, and H. Nakamura, “Estimation of left-turning vehicle maneuvers for the assessment of pedestrian safety at intersections,” ITSS research, vol. 36, no. 1, pp. 66–74, 2012.
[3] E.-H. Choi, “Crash factors in intersection-related crashes: An on-scene perspective,” Tech. Rep., 2010.
[4] A. Habibovic and J. Davidsson, “Requirements of a system to reduce car-to-vulnerable road user crashes in urban intersections,” Accident Analysis & Prevention, vol. 43, no. 4, pp. 1570–1580, 2011.
[5] M. S. Shirazi and B. T. Morris, “Looking at intersections: A survey of intersection monitoring, behavior and safety analysis of recent studies,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 1, pp. 4–24, 2016.
[6] U. Franke, D. Gavrilu, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler, “Autonomous driving goes downtown,” Intelligent Systems, no. 6, pp. 40–48, 1998.
[7] N. Saunier and T. Sayed, “Probabilistic framework for automated analysis of exposure to road collisions,” Transportation Research Record, 20083, no. 1, pp. 96–108, 2008.
[8] Á. Svensson and C. Hydén, “Estimating the severity of safety related behaviour,” Accident Analysis & Prevention, vol. 38, no. 2, pp. 379–385, 2006.
[9] T. Sayed and S. Zein, “Traffic conflict standards for intersections,” Transportation Planning and Technology, vol. 22, no. 4, pp. 309–323, 1999.
[10] S. R. Perkins and J. L. Harris, “Traffic conflict characteristics-accident potential at intersections,” Highway Research Record, no. 225, 1968.
[11] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” in In Proceedings of Advances in Neural Information Processing Systems (NIPS), 2015, pp. 3483–3491.
[12] H. Cheng, H. Liu, T. Hirayama, F. Shimamura, N. Akai, and H. Murase, “Automatic interaction detection between vehicles and vulnerable road users during turning at an intersection,” in Proceedings of the 31st IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, vol. 19, 2020.
[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in In Proceedings of Advances in Neural Information Processing Systems (NIPS), 2017, pp. 5998–6008.
[14] B. L. Allen, B. T. Shin, and P. J. Cooper, “Analysis of traffic conflicts and collision,” Transportation Research Record, vol. 677, pp. 67–74, 1978.
[15] R. Compton and E. Milton, “Safety impact of permitting right-turn-on-red,” National Highway Traffic Safety Administration, Report No. DOTHS-808, 1994.
[16] I. Kaparias, M. G. Bell, J. Greensted, S. Cheng, A. Miri, C. Taylor, and B. Mount, “Development and implementation of a vehicle–pedestrian conflict analysis method: adaptation of a vehicle–vehicle technique,” Transportation Research Record, vol. 2198, no. 1, pp. 75–82, 2010.
[17] L. Salamati, B. Schroeder, N. M. Roupaffi, C. Cunningham, R. Long, and J. Barlow, “Development and implementation of conflict-based assessment of pedestrian safety to evaluate accessibility of complex intersections,” Transportation research record, vol. 2264, no. 1, pp. 148–155, 2011.
[18] T. Sayed, M. H. Zaki, and J. Autey, “Automated safety diagnosis of vehicle–bicycle interactions using computer vision analysis,” Safety science, vol. 59, pp. 163–172, 2013.
[19] K. Ismail, T. Sayed, N. Saunier, and C. Lim, “Automated analysis of pedestrian–vehicle conflicts using video data,” Transportation Research Record, vol. 2140, no. 1, pp. 44–54, 2009.
[20] K. Ismail, T. Sayed, and N. Saunier, “Automated analysis of pedestrian–vehicle conflicts: Context for before-and-after studies,” Transportation Research Record, vol. 2198, no. 1, pp. 52–64, 2010.
[21] I. Kaparias, M. G. Bell, W. Dong, A. Sastrawinata, A. Singh, X. Wang, and B. Mount, “Analysis of pedestrian–vehicle traffic conflicts in street designs with elements of shared space,” Transportation research record, vol. 2393, no. 1, pp. 21–30, 2013.
[22] A. Tageldin and T. Sayed, “Developing evasive action-based indicators for identifying pedestrian conflicts in less organized traffic environments,” Journal of Advanced Transportation, vol. 50, no. 6, pp. 1193–1208, 2016.
[23] Y. Ni, M. Wang, J. Sun, and K. Li, “Evaluation of pedestrian safety at intersections: A theoretical framework based on pedestrian-vehicle interaction patterns,” Accident Analysis & Prevention, vol. 96, pp. 118–129, 2016.
[24] J. C. Hayward, “Near-miss determination through use of a scale of danger,” Highway Research Record, no. 384, pp. 24–34, 1972.
[25] K. Suzuki and H. Nakamura, “Traffic analyzer-the integrated video image processing system for traffic flow analysis,” in Proceedings of the 13th ITS World Congress, London, 8-12 October 2006, 2006.
[26] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” Pattern Analysis and Machine Intelligence, vol. 26, no. 1, pp. 92–116, 2004.
[27] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, “Are they going to cross? A survey of pedestrian detection and collision,” Transportation Research Record, vol. 2183, no. 1, pp. 96–108, 2008.
[28] A. Chorom, N. Mackowick, M. Pantic, N. Beuter, L. Drumond, F. Diego, and B. Ommer, “Learning to forecast pedestrian intention from pose dynamics,” in 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018, pp. 1277–1284.
D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in 2015 International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2015, pp. 3132–3137.

A. Rasouli and J. K. Tsotsos, “Autonomous vehicles that interact with pedestrians: A survey of theory and practice,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 900–918, 2019.

S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2016, pp. 779–788.

Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, “M2det: A single-shot object detector based on multi-level feature pyramid network,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 9259–9266.

B. K. Horn and B. G. Schunck, “Determining optical flow,” Artificial intelligence, vol. 17, no. 1-3, pp. 185–203, 1981.

G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in Scandinavian conference on Image analysis. Springer, 2003, pp. 363–370.

D. J. Rezende, S. Mohamed, and D. Wierstra, “Stochastic backpropagation and approximate inference in deep generative models,” in International conference on machine learning. PMLR, 2014, pp. 1278–1286.

D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014, pp. 3581–3589.

S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in International conference on machine learning. PMLR, 2015, pp. 448–456.

E. Parzen, “On estimation of a probability density function and mode,” The annals of mathematical statistics, vol. 33, no. 3, pp. 1065–1076, 1962.

D. O. Loftsgaarden, C. P. Quesenberry et al., “A nonparametric estimate of a multivariate density function,” The Annals of Mathematical Statistics, vol. 36, no. 3, pp. 1049–1051, 1965.

C. Koetsier, S. Busch, and M. Sester, “Trajectory extraction for analysis of unsafe driving behaviour,” International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 2019.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2015.

Hai lòng LIU (Member, IEEE) received the M.Eng. and the Ph.D. degrees in Engineering Science from Osaka University in 2002 and 2005, respectively. From 2005 to 2011, he was a Research Assistant Professor at the Graduate School of Informatics, Kyoto University. In 2011, he moved to the Graduate School of Information Science, Nagoya University as an Assistant Professor from 2012 to 2014, and a Designated Associate Professor from 2014 to 2017. He is currently a Designated Associate Professor at the Institutes of Innovation for Future Society, Nagoya University. His research interests include computer vision and human-computer interaction. He is a member of IEEE, ACM, IEICE, and IPSJ.

Hiroshi Murase received his B.Eng., M.Eng., and Ph.D. degrees in Electrical Engineering from Nagoya University, Japan. In 1980 he joined the Nippon Telegraph and Telephone Corporation (NTT). From 1992 to 1993, he was a visiting research scientist at Columbia University, New York. From 2003 he is a professor of Nagoya University, Japan. He was awarded the IEICE Shinohara Award in 1986, the Telecom System Award in 1992, the IEEE CVPR (Conference on Computer Vision and Pattern Recognition) Best Paper Award in 1994, the IPS Japan Yamashita Award in 1995, the IEEE ICRA (International Conference on Robotics and Automation) Best Video Award in 1996, the Takayanagi Memorial Award in 2001, the IEICE Achievement Award in 2002, and the Ministry Award from the Ministry of Education, Culture, Sports, Science and Technology in 2003. He is a Fellow of IEEE, IEICE, and IPSJ Japan.

Takatsugu Hirayama (Member, IEEE) received the M.Eng. and Ph.D. degrees in Engineering Science from Osaka University in 2002 and 2005, respectively. From 2005 to 2011, he was a Research Assistant Professor at the Graduate School of Informatics, Kyoto University. In 2011, he moved to the Graduate School of Information Science, Nagoya University as an Assistant Professor from 2012 to 2014, and a Designated Associate Professor from 2014 to 2017. He is currently a Designated Associate Professor at the Institutes of Innovation for Future Society, Nagoya University. His research interests include computer vision and human-computer interaction. He is a member of IEEE, ACM, IEICE, and IPSJ.

Hao Cheng received his B.A. degree in Information Management and Information System from Huazhong Agricultural University, China, in 2011, and his M.Sc. degree in Internet Technologies and Information Systems from Leibniz University Hannover, Germany, in 2017. He is currently finishing his Ph.D. at the Institute of Cartography and Geoinformatics at Leibniz University Hannover, Germany.

His research interests are deep learning and computer vision for road user behavior modeling in intelligent transport systems and autonomous driving.

Li Feng received her B.Sc. Degree in Communication Engineering from Xidian University, China, in 2016, and her M.Sc. Degree in Electrical Engineering and Computer Science from Leibniz University Hannover, Germany, in 2020.

She has been working as a deep learning algorithm engineer in China since graduation. Her main research interest is the visual task of lightweight object detection applied to Advanced Driver Assistance Systems.