CAIT-UTC-REG46

Driving behavioral learning leveraging sensing information from Innovation Hub

FINAL REPORT
Aug 2022

Submitted by:

Sharon Di
Associate Professor

Zhaobin Mo
PhD Candidate

Columbia University
500 West 120th Street

Peter Jin
Associate Professor

Yufei Huang
PhD Candidate

Rutgers University
500 Bartholomew Road,96 Frelinghuysen Rd

External Project Manager
Solomon Caviness, Transportation Department Head
Middlesex County
75 Bayard Street New Brunswick, NJ 08901

In cooperation with

Rutgers, The State University of New Jersey
And
State of New Jersey
Department of Transportation
And
U.S. Department of Transportation
Federal Highway Administration

Disclaimer Statement

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

The Center for Advanced Infrastructure and Transportation (CAIT) is a Regional UTC Consortium led by Rutgers, The State University. Members of the consortium are Atlantic Cape Community College, Columbia University, Cornell University, New Jersey Institute of Technology, Polytechnic University of Puerto Rico, Princeton University, Rowan University, SUNY - Farmingdale State College, and SUNY - University at Buffalo. The Center is funded by the U.S. Department of Transportation.

1. Report No.
CAIT-UTC-REG46

2. Government Accession No.

3. Recipient’s Catalog No.

4. Title and Subtitle

Driving behavioral learning leveraging sensing information from Innovation Hub

5. Report Date
Aug 2022

6. Performing Organization Code
CAIT/Columbia University

TECHNICAL REPORT STANDARD TITLE PAGE

7. Author(s)
Xuan (Sharon) Di https://orcid.org/0000-0003-2925-7697
Peter Jin https://orcid.org/0000-0002-7688-3730
Yufei Huang https://orcid.org/0000-0002-9996-9190
Zhaobin Mo https://orcid.org/0000-0002-0465-8550

8. Performing Organization Report No.
CAIT-UTC-REG46

9. Performing Organization Name and Address
Center for Advanced Infrastructure and Transportation
Rutgers, The State University of New Jersey 100 Brett Road,
Piscataway, NJ 08854

10. Work Unit No.

11. Contract or Grant No.
69A3551847102

12. Sponsoring Agency Name and Address
Center for Advanced Infrastructure and Transportation
Rutgers, The State University of New Jersey
100 Brett Road
Piscataway, NJ 08854

13. Type of Report and Period Covered
Final Report
1/1/2021 – 6/30/2022

14. Sponsoring Agency Code

15. Supplementary Notes
U.S. Department of Transportation/OST-R
1200 New Jersey Avenue, SE
Washington, DC 20590-0001

16. Abstract

With the accelerated deployment of connected and automated vehicle (CAV) technologies, public agencies have urgent needs on how to utilize these rich data sources of CAVs to improve traffic mobility, safety, and environmental and energy impact. This research will tackle one of the big data challenges, which is mining driving behavior patterns using vehicle data sources. We leverage physics-informed deep learning and uncertainty quantification methods to predict drivers’ car-following behavior using historical trajectories. A digital twin is developed leveraging the COSMOS testbed deployed near Columbia campus to validate the model algorithms and results. Moreover, an app is developed that captures drivers' faces. On the AWS server, face detection algorithms are applied to analyze drivers' moods and attention. Combined with the vehicle information (e.g., speed, acceleration) that is detected from roadside cameras, a model is established to predict the safety index of the driver and the roadway. The project outcome will be valuable for digital sibling simulation development and applications and future deployment of AVs that need to drive alongside humans.

17. Key Words
Driving behavior learning,
Physics-informed deep learning,
Face detection,
Safety index

18. Distribution Statement

19. Security Classification (of this report)
Unclassified

20. Security Classification (of this page)
Unclassified

21. No. of Pages
Total 27

22. Price

Acknowledgments

This project was financially supported by the Center for Advanced Infrastructure and Transportation (CAIT), a University Transportation Center (UTC) supported by USDOT-OST-R.

Table of Contents

List of Figures

List of Tab

1. Introduction and Problem Statement

Driving behavior has been extensively studied using various data sources, ranging from loop detectors, bluetooth, to GPS trajectories and mobile traces. Characterizing driving behavior is crucial to human-aware and social-aware motion planning of autonomous vehicles. However, understanding human driving behavior is challenging due to the intrinsic randomness of human behavior.

Risk assessment of roadways is usually practiced based on historical crash data, which ignores information of driver behaviors and real-time traffic situations. In this project, the Safe Route Mapping (SRM) model is extended by using both in-vehicle data and infrastructure-based data to predict safety index of the roadways.

2. Literature Review

2.1 Uncertainty Quantification using Physics-informed Deep Learning (PIDL)

The most widely used UQ methods include Bayesian approximation [1-5], ensemble methods [6,7], and generative models like the variational autoencoder [9,10] and generative adversarial networks (GAN) [11,12]. There is a growing trend in applying PIDL to UQ. One branch of studies apply physics-informed GANs (PhysGAN) to approximate solutions of partial differential equations (PDE) [13-15]. However, all these methods assume that the randomness arises from initial conditions or inputs while neglecting stochasticity in parameters associated with inherent physics or behaviors.

The other branch of studies apply PhysGAN to solve stochastic differential equations [16,17]. Although those studies assume that experimental data is generated from stochastic differential equations, they still use deterministic equations to calculate the physics discrepancy. Moreover, they demonstrate the results using only numerical data, and it remains a question whether those models can be applied to real-world cases. To this end, existing PhysGAN models may fail to capture the uncertainty arising from heterogeneity of drivers, which we believe is a major source of randomness when it comes to CF behavior prediction.

2.2 Driving Safety Index Prediction using both in-vehicle and roadside data

Safe Route Mapping (SRM) model was initially introduced by Jiang, etc.[18]. The Highway Safety Manual (HSM) published by the American Association of State Highway and Transportation Officials (AASHTO) outlines that developing Safety performance Functions (SPFs) can be used by jurisdictions to make better safety decisions [19]. The current practice of using SPF to predict average crash frequency suffers from a lack of sufficient data and model accuracy [20]. Naturalistic Driving Studies (NDS) reveals correlations between driver behaviors, roadway segments in either crash/near or normal situations [21]. However, the measurement of driving performance lacks driver behavior data because the information is hard to capture in real-time or requires additional devices inside the vehicle

In recent years, with the development of V2I technology and the lower cost of infrastructure sensing, collecting data inside a vehicle and from roadside infrastructure has become possible. Inside a vehicle, a driver concentration control system that monitors the driver's behavior mainly uses cameras, eye-trackers, or contactless sensors. [22] provides a solution that driver's performance can be captured using a smartphone application, which is more affordable compared to adding additional devices inside a vehicle. On the roadside, computer vision sensors based on traffic surveillance systems and cloud computing provides new opportunities for enhancing traffic safety. [23] introduced a computer vision algorithm to obtain vehicle trajectories from high-angle traffic video. Their model combines scanline-based trajectory extraction and feature-matching coordinate transformation to detect vehicles and get their traces. Their method has been proven to be robust and accurate.

3. Physics-informed Deep Learning For Car-following Modeling (Columbia)

3.1 Uncertainty Quantification of Car-following Behavior using GAN

UQ of CF behavior is illustrated in Figure 3.1. A red car is following a blue car along the horizontal axis, and the vertical axis is time. It is assumed that a driver obeys an underlying stochastic policy $\pi(a|s)$ that maps from driving states $s \in S$ to a distribution over actions $a \in A$. A CF model learns a surrogate policy $\pi_\theta(a|s)$ that approximates the ground-truth policy $\pi(a|s)$. At time step t, the red car samples its action a given its current state s, which leads to the true position (solid red car) at time step $t + \Delta t$. Meanwhile, a surrogate model π_θ predicts the action distribution and sample an action $\hat{a}$, which leads to the estimated position (transparent red car) at time step $t + \Delta t$. The key problem is to quantify the uncertainty of prediction $\hat{a}$ and its discrepancy with regards to the true action a.

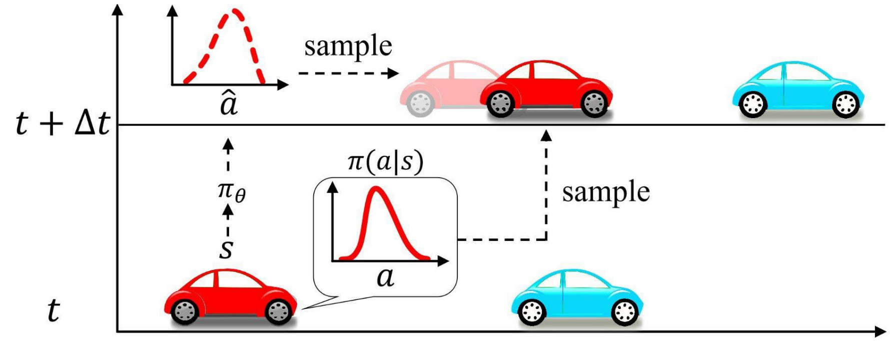


Figure 3.1: An illustration of UQ for CF behavior.

The structure of the DoubleGAN is illustrated in Figure 3.2. It consists of two parts: the left half (blue) contains the primal GAN and relevant variables; the right half (red) contains the auxiliary GAN and relevant variables. We will explain the left part first and then the right part. The primal GAN consists of the generator G_θ and the primal discriminator D_ϕ. Labeled states s_o are fed into the generator together with random noise z. The predicted state-action pairs $(s_o, \hat{a}_o)$ and the labeled state-action pairs (s_o, a_o) are judged by the primal discriminator D_ϕ, and the data loss is thus computed as $L_o(\theta) = \frac{1}{N_o}\sum_{i=1}^{N_o} \quad D_\phi(s_o^{(i)}, \hat{a}_o^{(i)})$.

The auxiliary GAN consists of the generator G_θ and the auxiliary discriminator D'_η. On one hand, unlabeled states s_c and random noise z are fed into the physics equation to generate physics predictions $\tilde{a}_c$. On the other hand, s_c and z are fed into the generator to get predictions $\hat{a}_c$. The auxiliary discriminator D'_η is trained to distinguish the generator-predicted state-action pairs $(s_c, \hat{a}_c)$ from the physics-predicted state-action pairs $(s_c, \tilde{a}_c)$, from which we can define the physics loss as$L_c(\theta, \lambda) = \frac{1}{N_c}\sum_{j=1}^{N_c} \quad D'_\eta(s_c^{(j)}, \hat{a}_c^{(j)})$.

L_c (θ,λ)=1/N_c ∑_(j=1)^(N_c) D_η^' (s_c^((j)),a^_c^((j))).

For simplicity, the loss functions of the discriminators D_ϕ and D'_η are not shown in Fig. 2.

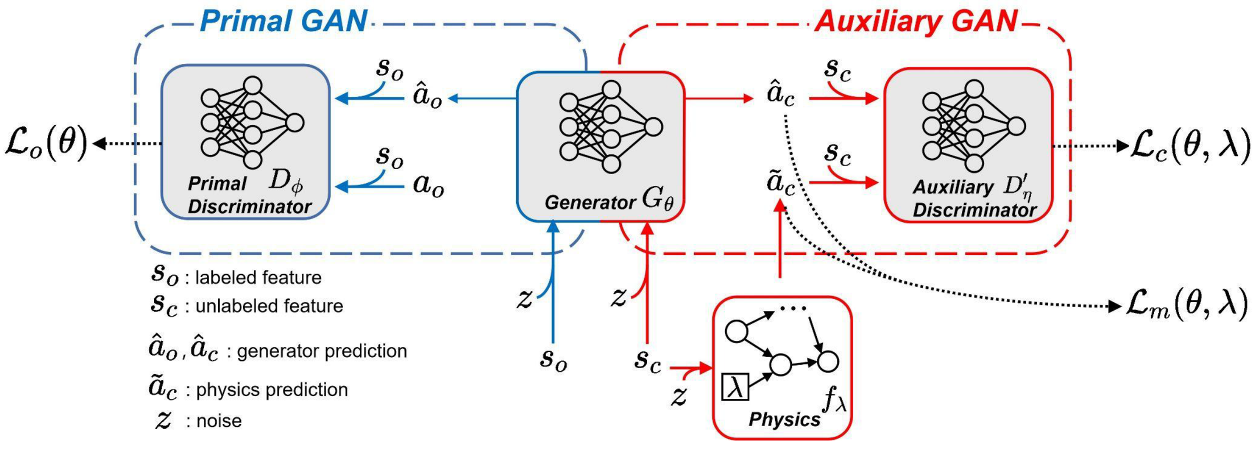

Primal GAN
Auxiliary GAN
$\mathcal{L}_o(\theta)$
s_o
$\hat{a}_o$
s_o
a_o
Primal Discriminator D_ϕ
Generator G_θ
$\hat{a}_c$
s_c
s_c
$\tilde{a}_c$
Auxiliary Discriminator D'_η
$\mathcal{L}_c(\theta, \lambda)$
$\mathcal{L}_m(\theta, \lambda)$
z
s_o
z
s_c
z
λ
Physics f_λ
s_o : labeled feature
s_c : unlabeled feature
$\hat{a}_o, \hat{a}_c$: generator prediction
$\tilde{a}_c$: physics prediction
z : noise

Figure 3.2 Structure of the DoubleGAN.

Although stochastic physics is encoded to better capture the real-world noise, an additional adversarial loss is included, which may hinder the model convergence. To tackle this problem, we propose to use the moment-matching technique to speed up the convergence of the DoubleGAN without incurring the model collapse. The moment-matching loss is depicted as:

$$L_m(\theta,\lambda) = \rho E_{q(s_c)}\left[\frac{\mu(\tilde{a}_c)-\mu(\hat{a}_c)}{|\mu(\tilde{a}_c)|+|\mu(\hat{a}_c)|}\right]^2 + (1-\rho)E_{q(s_c)}\left[\frac{\sigma(\tilde{a}_c)-\sigma(\hat{a}_c)}{|\sigma(\tilde{a}_c)|+|\sigma(\hat{a}_c)|}\right]^2,$$

where the first and the second terms measure the discrepancies of the mean and the standard deviation between the neural network predictions and the physics predictions, respectively. μ is the operator for the mean, and σ is the operator for the standard deviation. To mitigate the effect of the scale difference between the first and the second moments, the sum of the absolute values of each moment is added as a normalization term. $\rho \in [0,1]$ is the ratio of each constraint component.

The physics model is jointly trained along with other networks by minimizing the physics loss $L_c(\theta,\lambda)$ with regard to both generator parameters θ and physics parameters λ on the unlabeled data. We illustrate the joint estimation in Figure 3.3. The line colors are associated with different types of data: the blue for the labeled data and the red for the unlabeled data. The solid lines indicate how generator G_θ and physics f_λ are trained: the generator G_θ is trained by both the labeled data and the samples of the physics f_λ, and the physics f_λ is trained by the samples of the generator G_θ. The dashed line indicates that the physics could be pre-trained by the labeled data prior to the joint estimation.

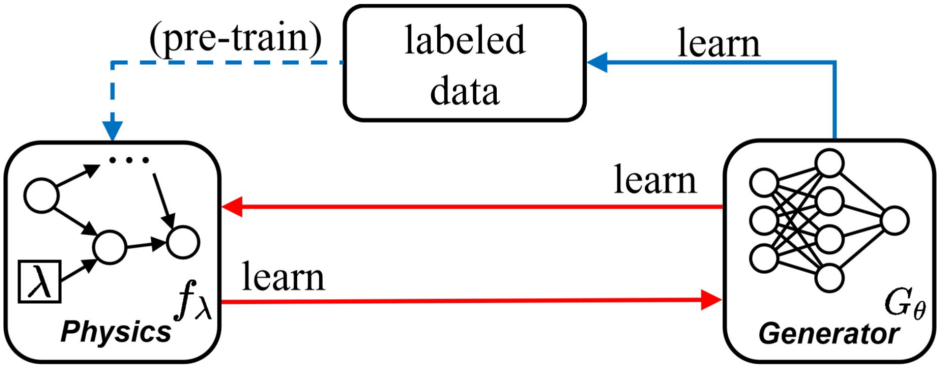


Figure 3.3: Joint estimation: training the physics model and the generator simutaneously.

The real-world data is from the Next Generation SIMulation (NGSIM) dataset, which is an open dataset that collects vehicle trajectories every 0.1 second. We focus on the US Highway 101.

We compare the prediction distribution of DoubleGAN with sample distribution at 4 randomly samples data points in Figu*re 3.4* (N_o = 500*). Most* parts of the predicted and ground-truth distributions overlap with each other, which demonstrates that DoubleGAN can capture the CF uncertainty of the real-world data well.

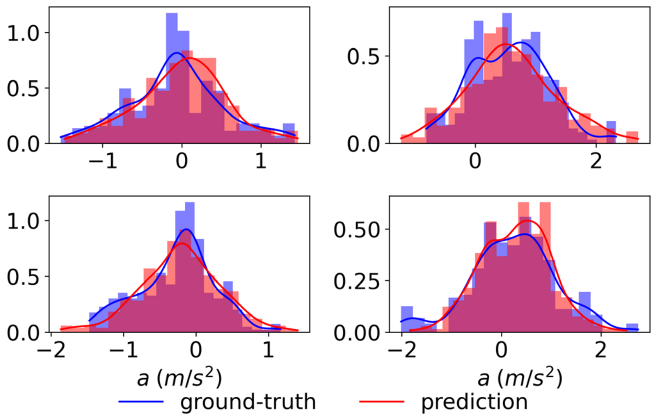
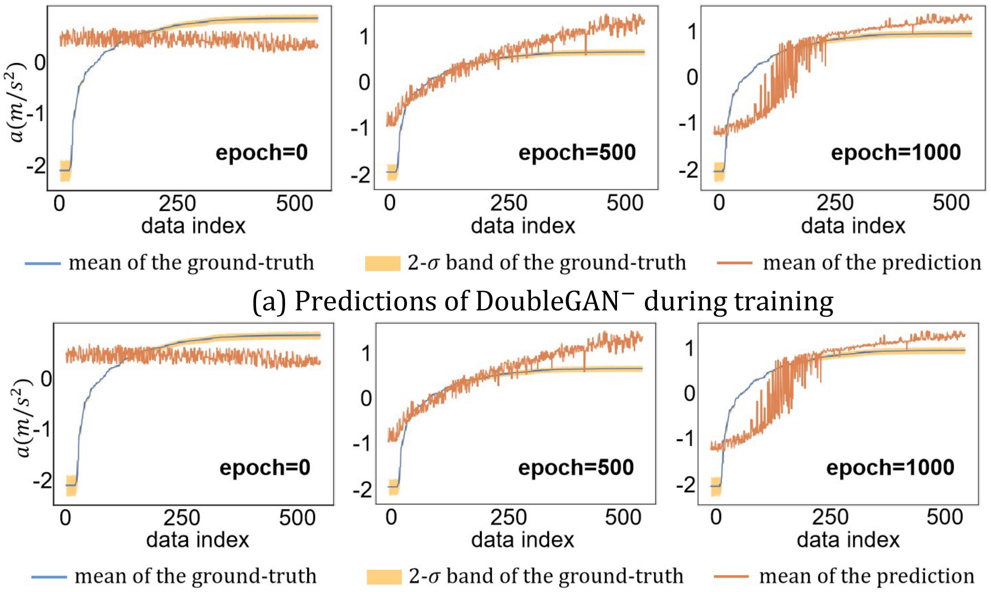


Figure 3.4: Visualization of predictions of DoubleGAN for the NGSIM data.

To evaluate the effectiveness of the moment-matching technique, we present the prediction results of the DoubleGAN$^-$ and DoubleGAN during the training process in Figure 3.5, which corresponds to the normal noise data case with the training size $N_o = 500$. The x-axis is the index of the training data points, which is sorted by the value of the acceleration. The y-axis is the acceleration. The blue and green lines are the mean of the ground-truth and the prediction, respectively; the yellow band is the two-standard ($2 - \sigma$ band) of the ground-truth. We can see that, imposed with the moment-matching, DoubleGAN converges much faster than DoubleGAN$^-$. The reason is that, the discriminators are not well-trained at the early-training stage and thus cannot supervise the generator very well. In comparison, moment-matching directly computes the moment discrepancy between the generator and the physics, which can assist with the training of the generator throughout the training process.



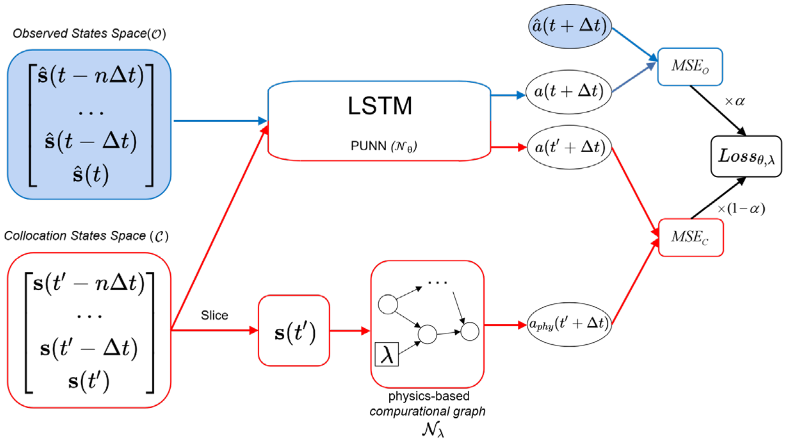
(a) Predictions of DoubleGAN$^-$ during training

(b) Predictions of DoubleGAN during training

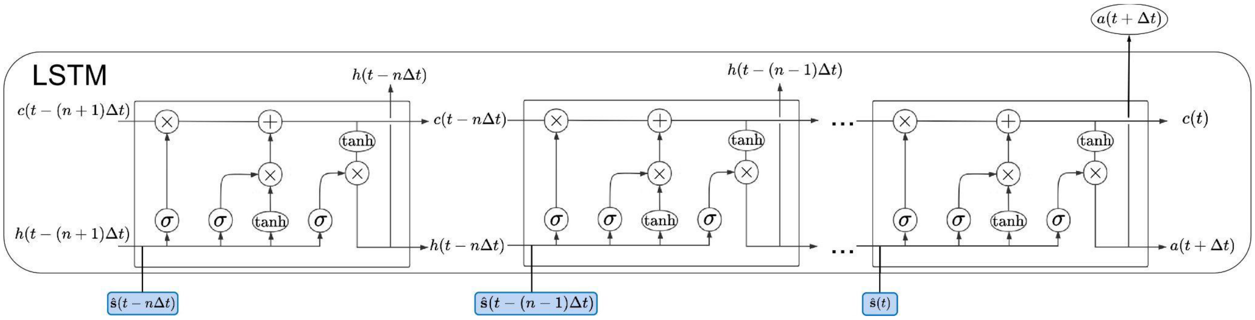
Figure 3.5: Predictions of DoubleGAN⁻ (top row) and DoubleGAN (bottom row) during the training process. The comparison between the model prediction and the ground-truth is presented at training epochs 0, 500, and 1000

3.2 Sequential prediction using LSTM

To validate the functionality of the PIDL method for other deep learning model, we changed the DL component of the PIDL from an ANN to a long short-term memory (LSTM) model. The LSTM model is one of the states-of-the-art DL model used in the car-following behavior modeling. Different from NNs, LSTM considers historic data when making a prediction. The PIDL-LSTM structure is shown in Figure 3.6. Figure 3.6 (a) introduces how to incorporate LSTM into the PIDL architecture, and Figure 3.6 (b) zooms into the LSTM component with the details of LSTM cells. In Figure 3.6 (a), $s(t)$ is the feature vector $[h(t), \Delta v(t), v(t)]$, n is the gap length. Note that there is no historic component in the physics-based CFM, only the latest feature is used for the physics regardless of the gap length. In Figure 3.6 (b), each rectangle encloses an LSTM cell. The row element of the input data, which is a feature vector, $\hat{s}(t)$ as the observed feature and $s(t')$ as the collocation feature, is fed into the LSTM cell sequentially. For each input feature $s(t)$, the LSTM cell outputs the cell state $(c(t))$ and hidden state $(h(t))$, which are also fed into the LSTM cell for the next time step. The hyperbolic tangent function, denoted as $tanh(\cdot)$, is a non-linear transform function, which adds non-linearity to the LSTM model. The sigmoid function, denoted as $\sigma(\cdot)$, outputs a number between zero and one, which controls how much information can pass through. One means letting everything through and zero means letting nothing through.



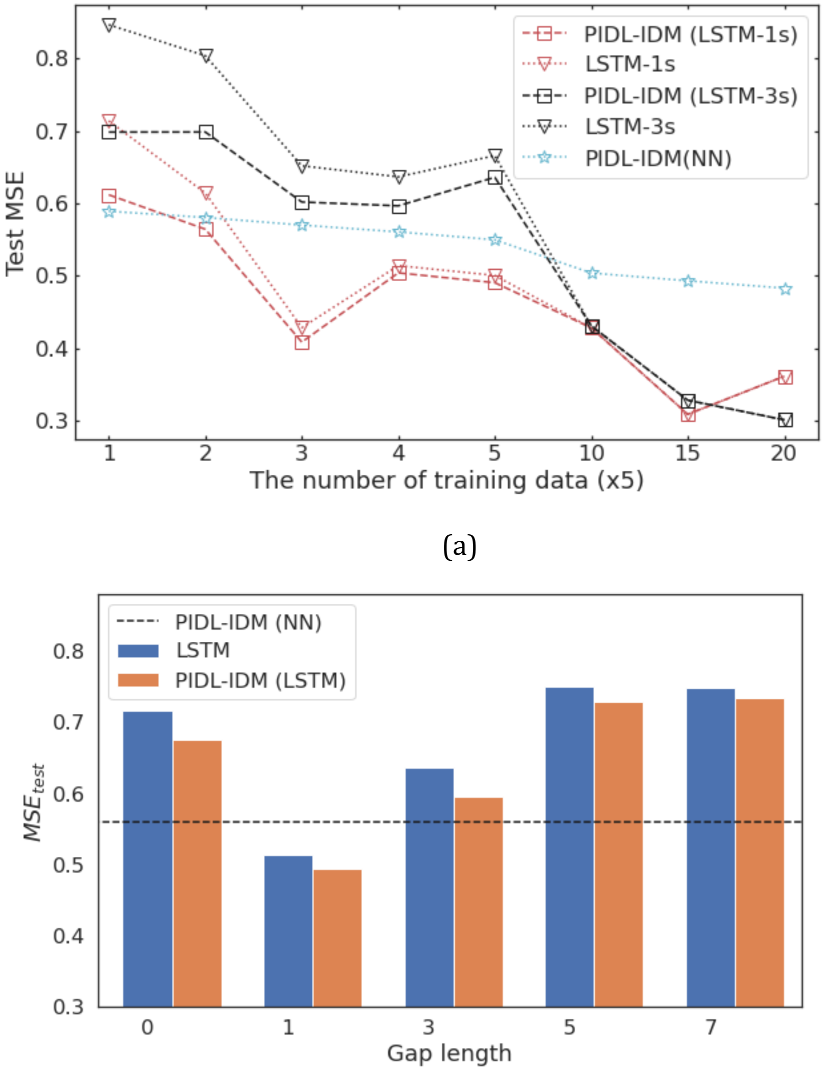
(a) Structure of PIDL-LSTM.



(b) LSTM cells. h and c are hidden states and cell states. $\sigma(\cdot)$ and $tanh(\cdot)$ denote the sigmoid and hyperbolic tangent functions, respectively.

Figure 3.6: PIDL-LSTM model.

We apply the LSTM-based PIDL-IDM to NGSIM and present results in Figure 3.7. Figure 3.7 (a) shows the test MSE on two LSTM-based PIDL-IDM models, one with a 1-second gap length (indicated by dashed red line with squares) and the other with a 3-second gap length (indicated by dashed black line with squares), against the training data size. Baseline models include pure data-driven LSTM with gap lengths to be 1 second (dashed red line with inverted triangles) and 3 seconds (dashed black line with inverted triangles), together with an ANN based PIDL-IDM model (dashed blue line with stars). The x-axis the training data size and the y-axis is the test MSE. When the length gap is 3 seconds, both the LSTM and the LSTM-based PIDL model are worse than the ANN-based PIDL model. The LSTM-based PIDL model achieves the best MSE when the gap length is 1 second, but it can only outperform the ANN-based PIDL model when the training data size is more than 10. The LSTM-based PIDL model show its advantage when the training data size is more than 25. It is because the LSTM's complex structure requires more training data but at the same time is more capable of learning driver reaction. Figure 3.7 (b) shows the PIDL-LSTM results across different gap lengths when the number of training data is fixed as 20. Note that when the gap length equals to 0, it means that LSTM just uses data of time step t as how the NN does. We can see that the LSTM-based PIDL can only outperform the ANN-based PIDL when the gap length equals to 1 second, this is partially because the LSTM-based PIDL model of longer gap lengths demand more training data to achieve a similar performance level as that of shorter gap lengths, and thus may not be amenable when we have sparse observations. In conclusion, the PIDL with more complex NN architecture requires more data to train and may not outperform those with simpler NN architecture, especially when data is insufficient. When the training data grows, LSTM-PIDL is superior to the ANN-based PIDL.



(a)

(b)

Figure 3.7: Results of PIDL-IDM (LSTM) with varying numbers of training data (a) and varying gap lengths (b).

4. Algorithm validation in a simulated environment (Columbia)

4.1 COSMOS Testbed

The COSMOS testbed is aimed at design, development, and deployment of a city-scale advanced wireless testbed in order to support real-world experimentation on next-generation wireless technologies and applications.

The COSMOS architecture has a particular focus on ultra-high bandwidth and low latency wireless communication tightly coupled with edge cloud computing. The COSMOS testbed will be deployed in upper Manhattan and will consist of 40-50 advanced software-defined radio nodes along with fiber-optic front-haul and back-haul networks and edge and core cloud computing infrastructure. Researchers will be able to run experiments remotely on the COSMOS testbed by logging into a web-based portal which will provide various facilities for experiment execution, measurements, and data collection.

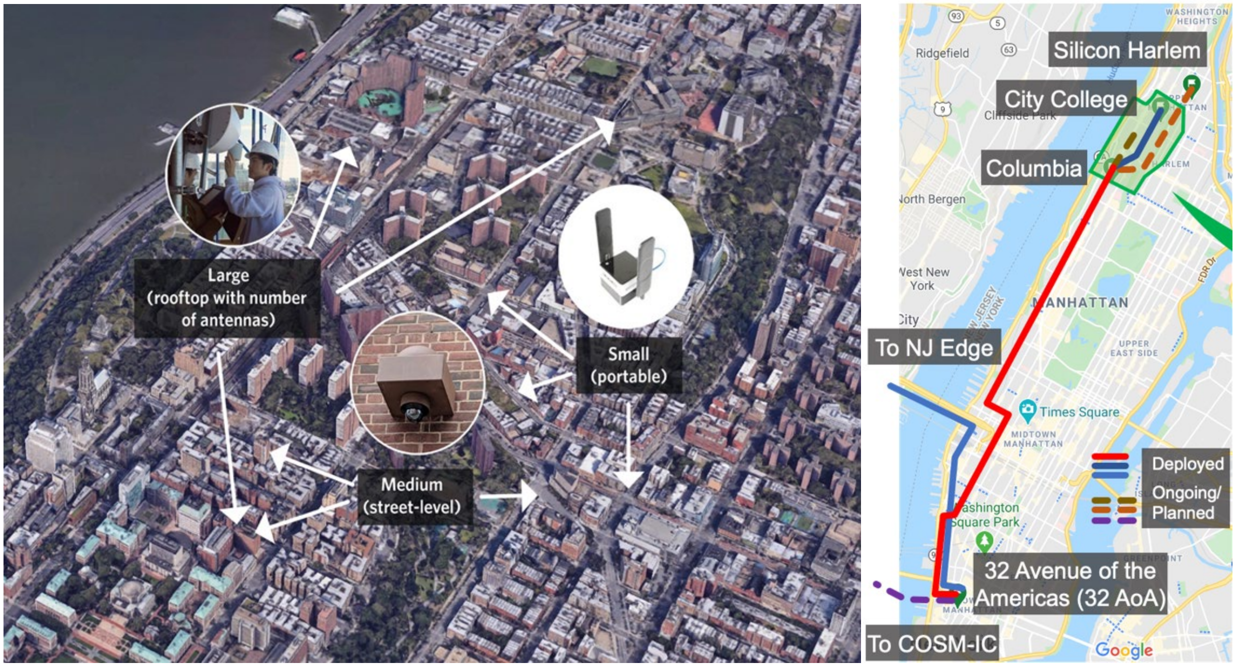


Figure 4.1: Hardware and location for COSMOS testbed.

4.2 Digital Twin

Based on the COSMOS testbed, we build a digital twin to validate our algorithms. With the help of digital twin, we can control an individual vehicle or gather needed information from the world. It's flexible to deploy our algorithms and do testing.

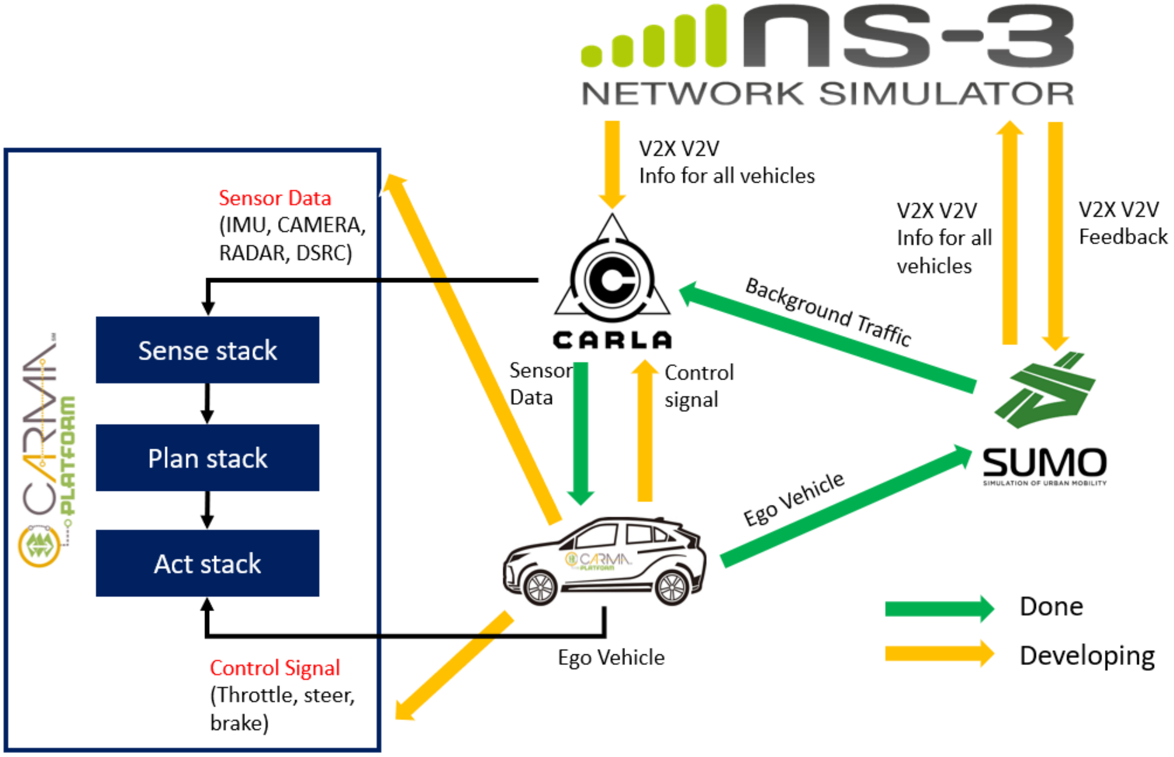


Figure 4.2: Structure of the digital twin.

For the digital twin, we set up a simulation environment mainly using CARLA and SUMO, two popular open-source simulators. CARLA has been developed from the ground up to support development, training, and validation of autonomous driving systems. And it supports a wide range of specification of sensor suites, environmental conditions, full control of all actors and map generation. SUMO allows addressing a large set of traffic management topics. It is purely microscopic: each vehicle is modeled explicitly, has its own route, and moves individually through the network. And we use NS-3 to simulate the communication module in digital twin, which is a discrete-event network simulator for Internet systems. By setting up these environments, we are able to spawn an ego vehicle and deploy different algorithms flexibly.

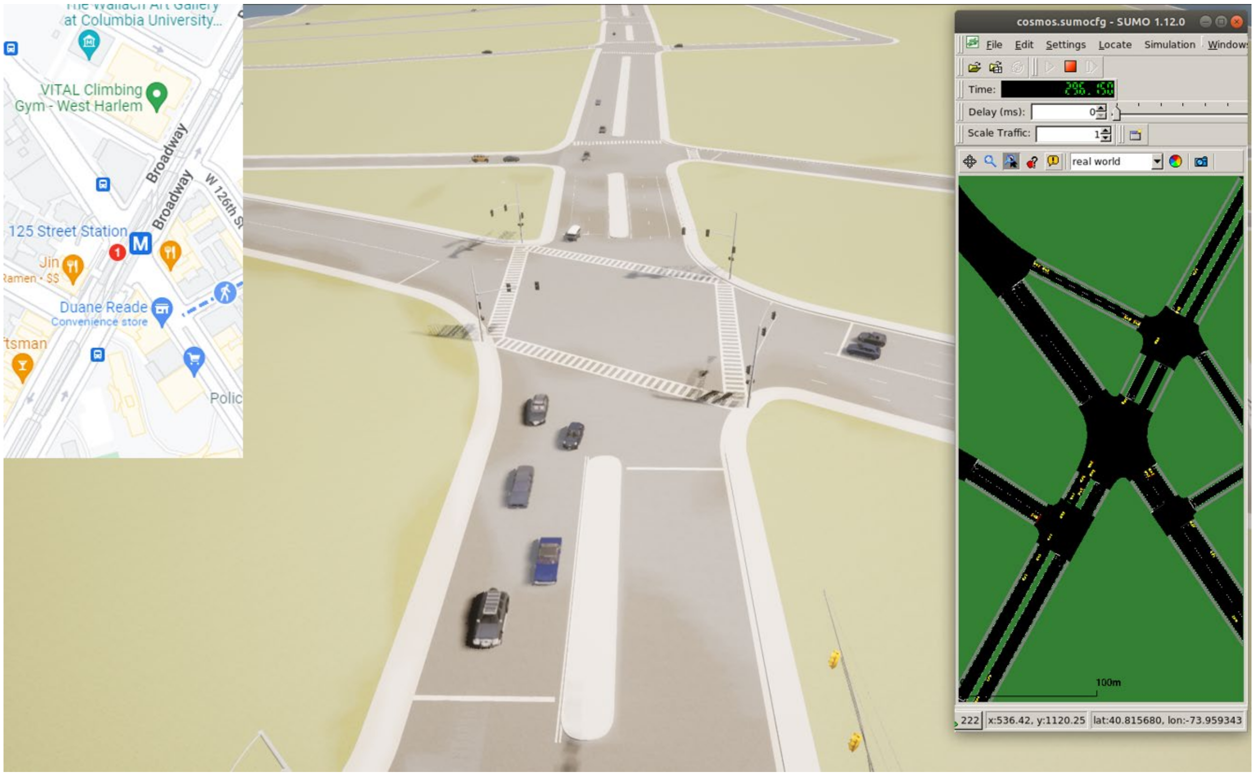

The Wallach Art Gallery at Columbia University...
VITAL Climbing Gym - West Harlem
Broadway
Broadway
W 126th St
125 Street Station
M
1
Jin
Ramen · $$
Duane Reade
Convenience store
tsman
Polic
cosmos.sumocfg - SUMO 1.12.0
File Edit Settings Locate Simulation Window
Time: 296.150
Delay (ms): 0
Scale Traffic: 1
real world
100m
222 x:536.42, y:1120.25 lat:40.815680, lon:-73.959343

Figure 4.3: CARLA-SUMO co-simulation.

By using digital twin, we could replicate a real car-following scenario. As the video shows, we first preprocess the real world data from the labeled lidar points from Waymo open dataset and replicate the scenario in CARLA simulator. Vehicles have exact relevant positions in the simulator environment, so it is convenient to regenerate the lidar points or camera images with the replicated scenario in CARLA. And it is possible to test our algorithms using exact real world data and even augment the data.

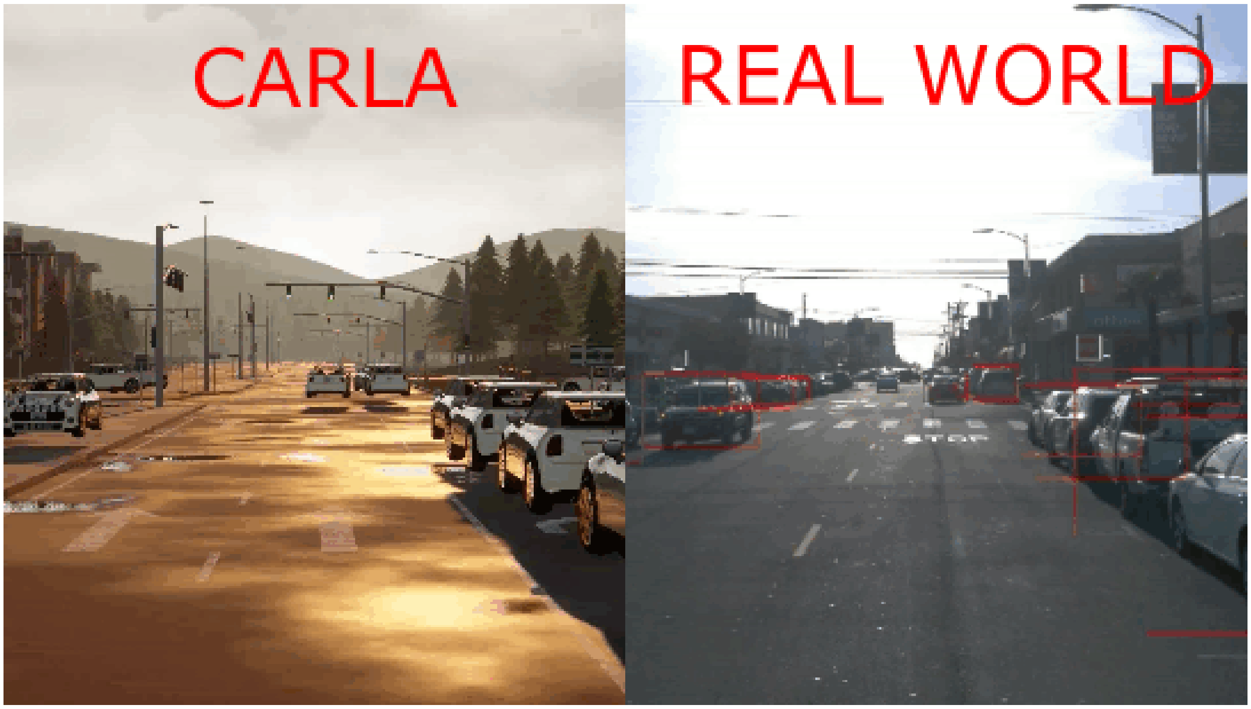


Figure 4.4: Comparing video generated in CARLA and real world.

5. Driving Safety Index Prediction using both in-vehicle and roadside data

5.1 Collecting driver's information

In order to capture real-time driver behaviors, an Android App is developed to gather information from the driver inside a vehicle and upload them to a server. An Android phone is placed on the dashboard inside a car and the driver's face is captured using its front camera. Figure 5.1 illustrates the user interface of the APP.

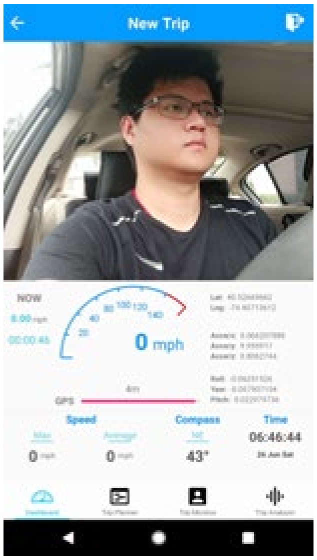


Figure 5.1: User Interface of the APP.

On the server, the driver's video is processed using the OpenFace toolkit to recognize the face and extract the facial landmarks. OpenFace is an open-source facial behavior analysis toolkit that is capable of detecting facial landmarks, estimate head pose and eye-gaze directions as is shown in Figure 5.2.

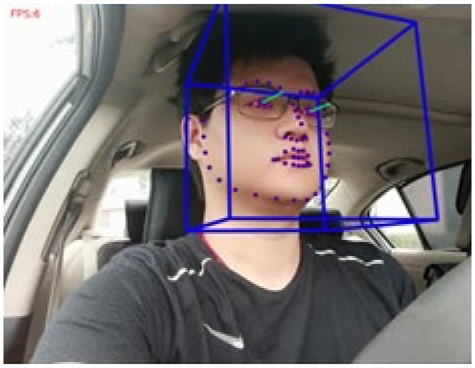


Figure 5.2: Facial recognition based on OpenFace toolkit.

Using the facial landmarks around the driver's eyes, Eye Aspect Ratio r_e (EAR) can be calculated using equation below. EAR is commonly used for fatigue level classification. If eye-blinking frequency radically increases, we would observe continuous changes of EAR. The threshold of r_e is set at 0.26 based on an experiment shown in Figure. 5.3. The eyes are considered closed when is smaller than the threshold.

$$r_e = \frac{\sqrt{(x_2-x_6)^2+(y_2-y_6)^2}+\sqrt{(x_3-x_5)^2+(y_3-y_5)^2}}{2\cdot\sqrt{(x_4-x_1)^2+(y_4-y_1)^2}}$$

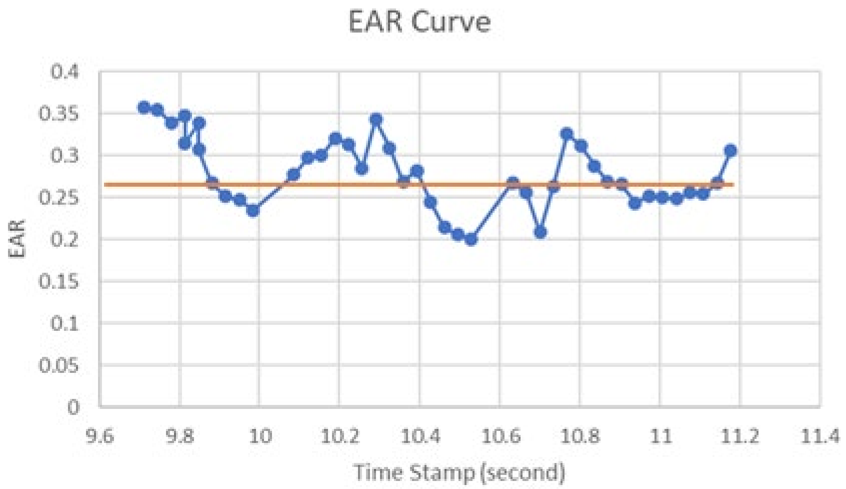


Figure 5.3: EAR curve experiment results

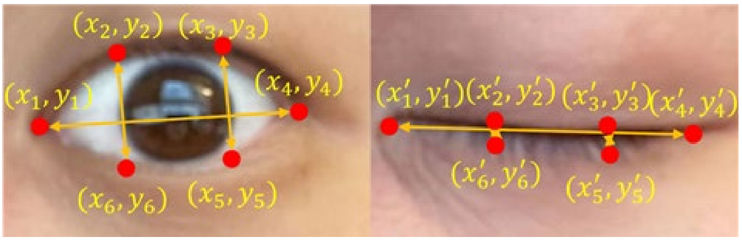

(x_2, y_2) (x_3, y_3)
(x_1, y_1)
(x_4, y_4)
(x_6, y_6) (x_5, y_5)
$(x'_1, y'_1)(x'_2, y'_2)$ $(x'_3, y'_3)(x'_4, y'_4)$
(x'_6, y'_6) (x'_5, y'_5)

Figure 5.4: Facial landmarks around the eyes

Given the eye-gaze directions, K-Nearest Neighbors algorithm (k-NN) [24] is applied to cluster the points that represent where the driver is looking at using the focusing area coordinates . As is shown in Figure 5.4, the pre-collected focusing points are grouped into 3 clusters. Based on where each point is locating at, we can approximately understand whether the driver is looking straight ahead or not.

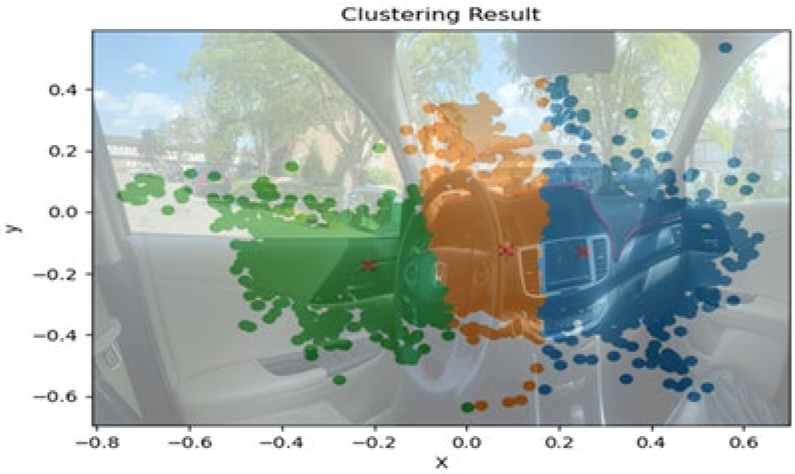


Figure 5.5: The approximation of driver's focusing area.

Besides drivers' focusing area and fatigue level, emotion also affects human performance through influencing individuals' judgment and behavior. For example, stressed operators could not achieve their optimal performance in complex task environments. Facial Action Coding System (FACS) is applied to figure out the driver's emotion. FACS refers to a set of facial muscle movements that correspond to a displayed emotion and can be recognized by the movement of facial landmark coordinates. The relation between driving performance and emotion can be found in [25].

5.2 Collecting vehicle information

Meanwhile, dynamic traffic information is captured by a roadside camera and uploaded to the same server. A longitudinal-scanline-based arterial traffic video analytics is applied to recognize vehicles from the video to build the speed and trajectory profiles [23]. We use three commonly used traffic conflict indicators (TCIs) to represent near misses [26], which are: (i) time to collision (TTC), (ii) modified time to collision (MTTC), and (iii) deceleration rate to avoid a crash (DRAC).

5.3 Driving risk score model and prediction results

Based on these profiles and driver's information, a LightGBM model is introduced to predict several traffic conflict indicators in the next one or two seconds. The structure of the model can be find in Figure 5.6.

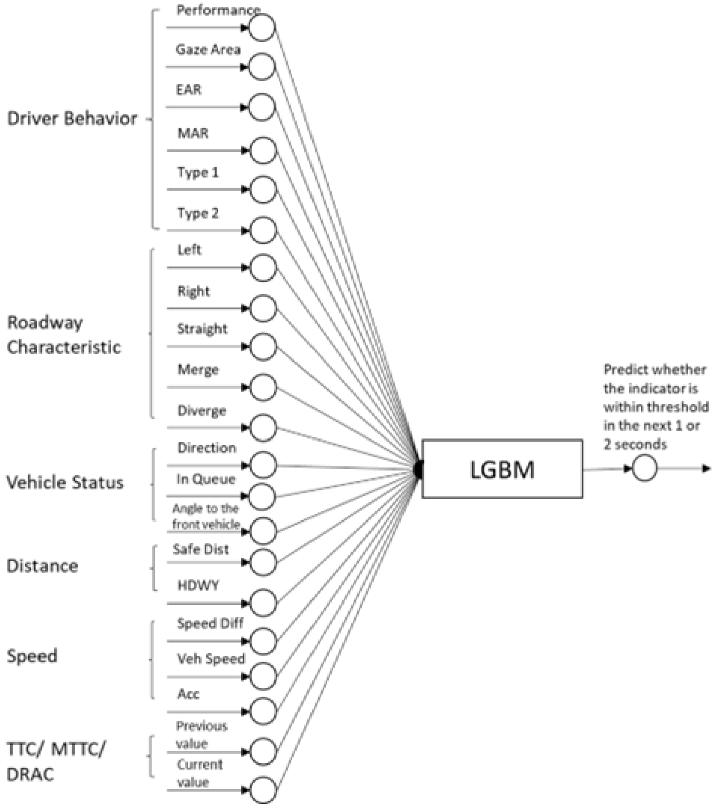
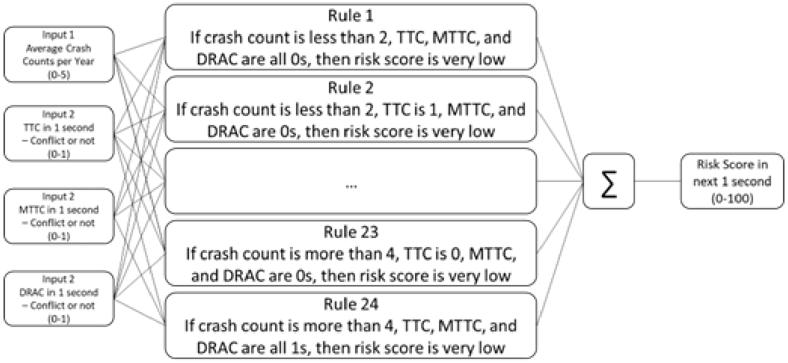


Figure 5.6: Structure of the LGBM.

Then, multiple data sources, including historical crash counts and predicted traffic conflict indicators of individual drivers, are combined using a fuzzy logic model to calculate risk scores for road segments as is shown in Figure 5.7.


Input 1
Average Crash Counts per Year
(0-5)
Input 2
TTC in 1 second – Conflict or not
(0-1)
Input 2
MTTC in 1 second – Conflict or not
(0-1)
Input 2
DRAC in 1 second – Conflict or not
(0-1)
...
∑
Risk Score in next 1 second
(0-100)

Rule 1
If crash count is less than 2, TTC, MTTC, and DRAC are all 0s, then risk score is very low

Rule 2
If crash count is less than 2, TTC is 1, MTTC, and DRAC are 0s, then risk score is very low

Rule 23
If crash count is more than 4, TTC is 0, MTTC, and DRAC are 0s, then risk score is very low

Rule 24
If crash count is more than 4, TTC, MTTC, and DRAC are all 1s, then risk score is very low

Figure 5.7: Structure of the Fuzzy Logic.

The proposed SRM model is illustrated for New Brunswick Smart Intersection Mobility Testbed shown in Figure 5.8. Real traffic data is collected and Table x shows the confusion matrix for the risk score levels.

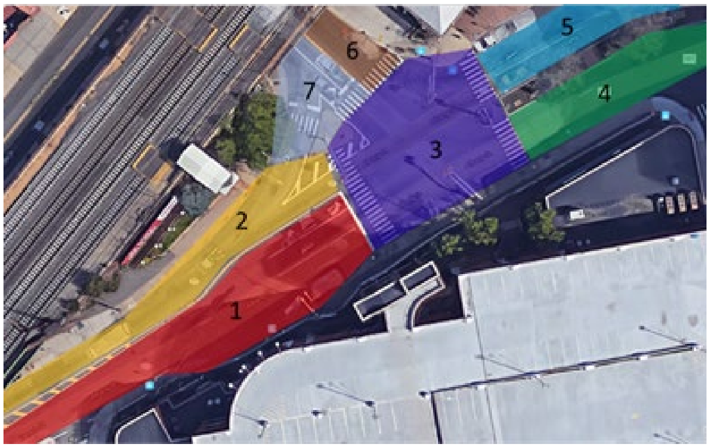
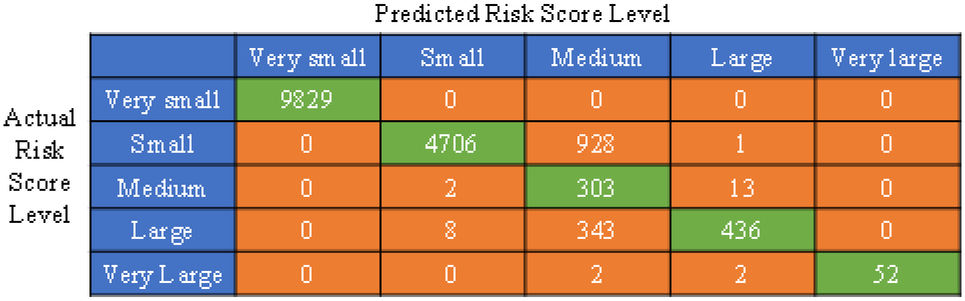


Figure 5.8: Illustration of road segments in New Brunswick Smart Intersection Mobility Testbed.

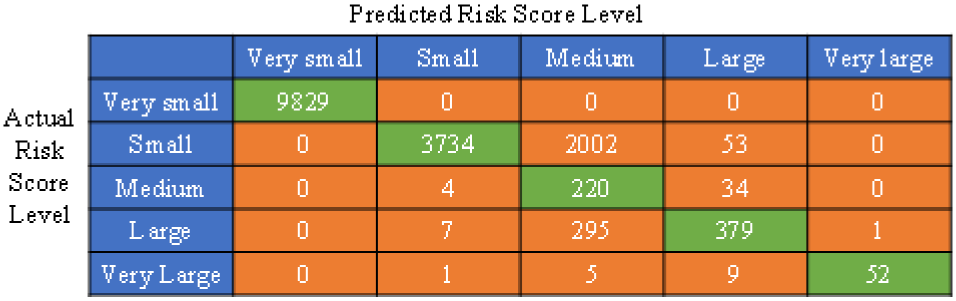
Table 5.1: Confusion Matrix for Risk Score Level (prediction for the next one second).



Predicted Risk Score Level

		Very small	Small	Medium	Large	Very large
Actual Risk Score Level	Very small	9829	0	0	0	0
	Small	0	4706	928	1	0
	Medium	0	2	303	13	0
	Large	0	8	343	436	0
	Very Large	0	0	2	2	52

Table 5.2: Confusion Matrix for Risk Score Level (prediction for the next two seconds).



Predicted Risk Score Level

		Very small	Small	Medium	Large	Very large
Actual Risk Score Level	Very small	9829	0	0	0	0
	Small	0	3734	2002	53	0
	Medium	0	4	220	34	0
	Large	0	7	295	379	1
	Very Large	0	1	5	9	52

Finally, risk heat maps are generated for visualization purposes as is shown in Figure 5.9. The authorities can use the dynamic heat map to designate safe corridors and dispatch law enforcement and drivers for early warning and trip planning.

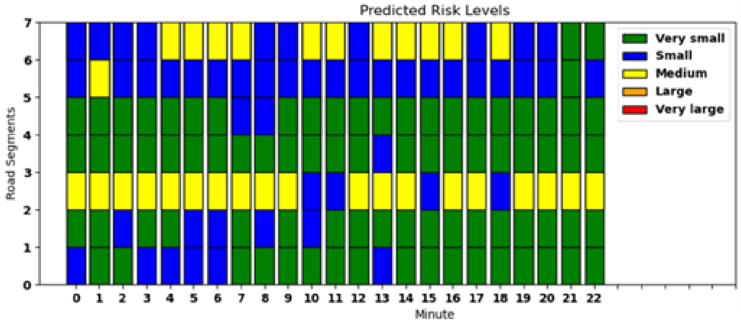


Figure 5.9: Risk heat map through time.

6. Conclusions and Recommendations

We propose a novel method called DoubleGAN that quantifies the uncertainty in human driver car-following (CF) behavior. The model encodes the stochastic physics information into the physics-informed generative adversarial network (PhysGAN) without incurring mode collapse. Using numerical data, we evaluate the performance of DoubleGAN under different noise types and training sizes. We further investigate the performance of DoubleGAN on a real-world dataset, the NGSIM dataset, and demonstrate that it outperforms baseline methods under different training sizes. Through ablation studies, we confirm that the moment-matching technique can speed up the model convergence and thus achieve better performance. By comparing DoubleGAN-IDM to DoubleGAN-Helly, we show that the most suitable physics model for DoubleGAN may not necessarily be the one with the best performance. The simplicity of physics is also a consideration. This work can be further improved in two directions. First, apart from the weighted sum, other approaches to integrating the data loss, moment-matching loss, and reconstruction loss can be proposed. Second, this work can be extended to quantify the uncertainty in sequential behavior, e.g., uncertainty quantification of human driving trajectory prediction.

We developed a method to collect real-time traffic data from roadside and driver's behavior data inside a vehicle. The contribution is that this work brings multiple sources of data together, especially human factors, when predicting traffic conflicts for individual drivers and calculate safety scores for road segments. There are certain limitations that we can address in our future work. For instance, more data need to be collected to test the robustness of the model. Combine real-world data with simulation results might be an approach to solve the problem.

References

[1] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun. 2020. Hands-on Bayesian Neural Networks–a Tutorial for Deep Learning Users. arXiv preprint arXiv:2007.06823 (2020)

[2] Mizanur Rahman, Mashrur Chowdhury, Taufiquar Khan, and Parth Bhavsar. 2015. Improving the efficacy of car-following models with a new stochastic parameter estimation and calibration method. IEEE Transactions on Intelligent Transportation Systems 16, 5 (2015), 2687–2699.

[3] Chris P IJ van Hinsbergen, Hans WC van Lint, Serge P Hoogendoorn, and Henk J van Zuylen. 2009. Bayesian calibration of car-following models. IFAC Proceedings Volumes 42, 15 (2009), 91–97.

[4] Hao Wang and Dit-Yan Yeung. 2016. Towards Bayesian deep learning: A framework and some existing methods. IEEE Transactions on Knowledge and Data Engineering 28, 12 (2016), 3395–3408.

[5] Kuan-Chieh Wang, Paul Vicol, James Lucas, Li Gu, Roger Grosse, and Richard Zemel. 2018. Adversarial distillation of bayesian neural network posteriors. In International Conference on Machine Learning. PMLR, 5190–5199.

[6] Ruihan Hu, Qijun Huang, Sheng Chang, Hao Wang, and Jin He. 2019. The MBPEP: a deep ensemble pruning algorithm providing high quality uncertainty prediction. Applied Intelligence 49, 8 (2019), 2942–2955.

[7] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. Advances in Neural Information Processing Systems 30 (2017).

[8] Tim Pearce, Felix Leibfried, and Alexandra Brintrup. 2020. Uncertainty in neural networks: Approximately bayesian ensembling. In International conference on artificial intelligence and statistics. PMLR, 234–244.

[9] Vanessa Böhm, François Lanusse, and Uroš Seljak. 2019. Uncertainty quantification with generative models. arXiv preprint arXiv:1910.10046 (2019).

[10] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013).

[11] Dhruv V Patel and Assad A Oberai. 2020. GAN-based Priors for Quantifying Uncertainty. arXiv preprint arXiv:2003.12597 (2020).

[12] Vinicius LS Silva, Claire E Heaney, and Christopher C Pain. 2021. GAN for time series prediction, data assimilation and uncertainty quantification. arXiv preprint arXiv:2105.13859 (2021).

[13] Arka Daw, M. Maruf, and Anuj Karpatne. 2021. PID-GAN: A GAN Framework Based on a Physics-Informed Discriminator for Uncertainty Quantification with Physics. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2021), 237–247.

[14] Yibo Yang and Paris Perdikaris. 2019. Adversarial uncertainty quantification in physics-informed neural networks. J. Comput. Phys. 394 (2019), 136–152.

[15] Dongkun Zhang, Lu Lu, Ling Guo, and George Em Karniadakis. 2019. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. J. Comput. Phys. 397 (2019), 108850.

[16] Liu Yang, Sean Treichler, Thorsten Kurth, Keno Fischer, David Barajas-Solano, Josh Romero, Valentin Churavy, Alexandre Tartakovsky, Michael Houston, Mr Prabhat, et al . 2019. Highly-scalable, physics-informed GANs for learning solutions of stochastic PDEs. In 2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS). IEEE, 1–11.

[17] Liu Yang, Dongkun Zhang, and George Em Karniadakis. 2020. Physics-informed generative adversarial networks for stochastic differential equations. SIAM Journal on Scientific Computing 42, 1 (2020), A292–A317.

[18] S. Jiang, M. Jafari, M. Kharbeche, M. Jalayer and K. N. Al-Khalifa, "Safe Route Mapping of Roadways Using Multiple Sourced Data," IEEE Transactions on Intelligent Transportation Systems, 2020.

[19] R. Srinivasan and K. Bauer, "Safety Performance Function Development Guide: Developing Jurisdiction-Specific SPFs," 2013.

[20] H. Huang, H. Zhou, J. Wang, F. Chang and M. Ma, "A multivariate spatial model of crash frequency by transportation modes for urban intersections," Analytic Methods in Accident Research, vol. 14, pp. 10-21, 2017.

[21] SWOV Institute for Road Safety Research, "Naturalistic Driving: Observing Everyday Driving Behavior," SWOV, The Hague, 2012.

[22] R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari and S. Jiang, "Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques," in The 9th International Conference on Ambient Systems, Networks, and Technologies (ANT), Porto, Portugal, 2018.

[23] T. T. Zhang, M. Guo, P. J. Jin, Y. Ge and J. Gong, "A longitudinal scanline based vehicle trajectory reconstruction method for high-angle traffic video," Transportation Research Part C Emerging Technologies, 2019.

[24] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," The American Statistician, vol. 46, no. 3, pp. 175-185, 1992.

[25] P. Madhavan, D. Wiegmann and F. Lacson, "Automation failures on tasks easily performed by operators undermine trust in automated aids," Human Factors, vol. 48, no. 2, pp. 241-256, 2006.

[26] G. Guido, F. Saccomanno, A. Vitale, V. Astarita and D. Festa, "Comparing Safety Performance Measures Obtained from Video Capture Data," Journal of transportation engineering, vol. 137, no. 7, 2011.