2025 IEEE 3rd International Conference on Mobility, Operations, Services and Technologies (MOST) | 979-8-3315-1160-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/MOST65065.2025.00024

Adaptive Dynamic Digital Twin for Test Scenario Generation

Hanlin Chen*, Guanhao Xu*, Anye Zhou*, Zejiang Wang†, Jieqiong Zhao‡, Shiwei Fang‡
**Oak Ridge National Laboratory*, Oak Ridge, TN, USA
Email: {chenh1, xug1, zhoua}@ornl.gov
†*University of Texas at Dallas*, Richardson, TX, USA
Email: zejiang.wang@utdallas.edu
‡*Augusta University*, Augusta, GA, USA
Email: {jiezhao, shfang}@augusta.edu

Abstract

***Abstract*—Vehicle testing has been an important part in the development of both highly automated vehicles (HAV) and advanced driving assistant systems (ADAS). Obtaining a good representation of the Vehicle Under Test (VUT) is crucial for test scenario library generation (TSLG). Current vehicle testing methods often involve calibrating car-following models using vehicle trajectory data to create static representations that cannot be dynamically updated. For instance, when multiple vehicle trajectories are collected, it is difficult to automatically determine whether a new trajectory improves the model's representativeness or degrades its accuracy.**

In this paper, we introduce a dynamically updated digital twin modeling framework featuring an adaptive mechanism that evaluates new trajectory data. This mechanism can decide whether to incorporate newly collected data into the current model or create a separate digital twin model when the trajectory significantly differs from prior data. Vehicle location, speed, and acceleration extracted from the newly collected trajectory data are used to support the dynamic update decision. By integrating this digital twin model into the test library generation process, we demonstrate its ability to assist in generating test libraries while effectively handling newly collected data.

***Index Terms*—Vehicle testing, digital twin, accelerated testing, test library generation**

I. Introduction

Vehicle testing has been an essential part in ensuring the safety, reliability, and performance of human-driven vehicles (HDVs), advanced driving assistant systems (ADASs) and highly automated vehicles (HAVs). Extensive research has been conducted, both theoretical [1] and practical [2], to advance this field. An efficient testing method is needed for vehicles, which are a typical type of cyber-physical safety-related system. To enhance efficiency and address ethical concerns, accelerated testing must be applied to obtain statistically significant estimation results before vehicle systems are deployed at scale in real world scenarios. Quite a lot of research has been done on accelerated testing for HAVs, and such a method can be migrated to ADASs. However, few studies focus on the accelerated testing for HDVs, and this is mainly due to the nature of human drivers. Compared to code-based driving principles for HAVs, HDVs differ from HAVs due to the black-box nature of the human driver's decision-making process. Various models have been proposed to explain or predict the decision process for human drivers. Most of them adhere to the universal stimulus-response framework [3], [4], yet the internal mechanisms of decision-making remain understudied compared to those of code-based HDVs or ADASs.

Common methods in accelerated testing include the generation of a test library, the selection of the surrogate vehicle under test (VUT) model, and the estimation of performance in VUT [5]. The selection of a surrogate model (SM) can have an impact on both library generation and performance evaluation. Current work on testing scenario library generation (TSLG) requires SM to calculate the criticality of the scenario, which subsequently affects the outcome of TSLG [6]. A significant challenge arises from the discrepancies between SM's predictions and actual VUT performance [7], potentially leading to errors in performance estimation. This issue is particularly pronounced in the performance estimation of HDVs, which exhibit greater variance compared to code- and computer-controlled HAVs [8]. Moreover, considering the variance in human drivers, the current static SMs deployed for TSLG may fail to account for the dynamic nature of human driver behavior as more HDV trajectories become available. To address those limitations, a better representation model of VUT is needed in the TSLG for HDVs, and in this paper, we propose to use a dynamically updated digital twin model as a representation of VUT in the TSLG problem.

Digital twins (DT) are the detailed digital replica of a component, product, or system that exists in the real world, and such a replica covers both the physical and functional aspects of the physical object, providing all potentially useful information for its current and future stages. This is the commonly accepted definition of DT by Boschert and Rosen [9]. Compared with pure simulation, the core difference lies within "Twin", which ensures a dynamic update mechanism so that the digital replica remains similar to the physical entity, as this concept

This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The publisher acknowledges the US government license to provide public access under the DOE Public Access Plan (https://energy.gov/downloads/doe-public-access-plan).

979-8-3315-1160-9/25/$31.00 ©2025 IEEE
DOI 10.1109/MOST65065.2025.00024

arises from the NASA Apollo project, according to the NASA technical report [10]. The dynamically updated property of the digital twin enables them as a good representation of physical property in the digital world. The post-damage investigation of the Apollo 13 spacecraft was the first application that utilizes the concept of a digital twin along with simulation to mirror the real-life conditions of the damaged spacecraft and enabled them to develop and test strategies to safely return astronauts to Earth [11]. Also, the close resemblance of a physical entity in the digital world makes them capable of performing multiple predictions in parallel, simultaneously. This serves as a great representation of VUT in TSLG as digital twins by default aim to close the gap of dissimilarity in the TSLG problem. However, not all trajectories should be taken into consideration when updating the parameters of the digital twin model, as the diversity in input data can greatly diminish the representativeness of the created digital twin model, if the diversity of the input data downgrades the centrality of the created model. This means that apart from the dynamic update mechanism within the digital twin model, a decision mechanism has to be implemented to determine if a newly collected trajectory can be used to update parameters or to create a new digital twin model for newly collected data [12]. Also, there exists a gap when a limited number of vehicle trajectories are collected and a new trajectory has to be classified as to which cluster it belongs or to initiate a new trajectory cluster for modeling. There are certain cases in which newly collected data were not seen anywhere. Therefore, it also remains a problem that in some cases we would need a digital twin model to react to an environment that was not seen in the dataset. Calibrated car-following models are commonly used in TSLG problem, but the dynamic update of a calibrated car-following model requires more trajectories to reach a new local optimum of the calibrated car-following model and thus will greatly decrease the update frequency for the digital replica of the VUT.

As stated above, there has not been significant research on performance evaluation regarding VUTs that are driven by humans, especially those assisted by AI systems. Part of the reason is the ethical dilemma inherent in testing and validating a countermeasure in an open driving environment. The consequence of an accident can result in severe physical and psychological harm to traffic participants, including the potential development of a long-lasting effect, namely post-traumatic stress disorder (PTSD). Research indicates that a significant portion of mother vehicle accident survivors experience psychological disorders. For instance, a study [13] found approximately 23% of the survivors reported psychological disorder after the accident, with 16. 5% still affected one year after the incident. Another study [14] found that even without severe physical harm or direct physical injury can suffer from PTSD. Therefore, even using a driving simulator for human participants, there is still a risk and potential mental harm to human participants if they experience accidents in the driving simulator, let alone the cost to the human subject. Thus, a good representation of the human driver is needed from the collected trajectories. Another reason for this is due to the diversity of human drivers, modeling of a human driver is harder compared to that with automated vehicles, and collecting human driver trajectories is also a problem due to the limited availability of HDV trajectories.

The high-level logic flow of this paper is shown in Figure 1. In this work, we propose a new digital twin creation framework that incorporates both passive vehicle response and the presentation of collective human-driver maneuvers. We also apply the acquired DT model in the downstream application for accelerated evaluation to show that our proposed model can be integrated as a dynamically updated digital replica of the physical entity and it is possible to utilize larger computational resources to accelerate the prediction of digital.

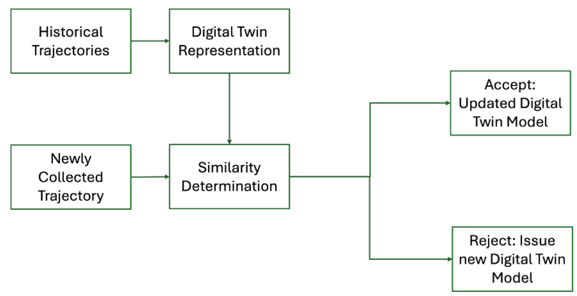

Fig. 1: Overall logic flow of proposed digital twin creation framework

The main contributions of the paper are listed below.

- We propose a framework for the creation of digital twin models and the updating of dynamic parameters for human drivers based on the modeling of historical trajectories and the imitation of learning through optimization.
- We propose a digital twin representation model that consideres both similarity to historical maneuver and the stimulate-and-response to the surrounding environment.
- We integrate dynamically updated digital twin representation of VUT into TSLG problem.

The paper is organized as follows. Section II reviews the relevant studies on the creation of digital twin models in transportation engineering and the evaluation of accelerated testing. Section III describes the modeling framework in our paper, including the high-level formulation of the problem, ego vehicle maneuver modeling based on historical data, and ego vehicle response based on model predictive control. Section IV presents a numerical case study for our proposed framework based on microscopic traffic simulations conducted in *Simulation of Urban Mobility* (SUMO) [15]. We showcased two distinct driving maneuvers and showed that our proposed framework can ensure the representativeness of the created digital twin with the integration of trajectory selection mechanism. Section V provides a discussion of the results, highlights the limitations, and concludes the entire work.

II. Related Works

A. Digital-Twin-Related Work in Transportation Engineering

Digital twin indicates a digital replica of the real-world entity. Following the definition in [10], the first attempt to use the digital twin as a prediction to assist optimization of system performance is founded in the investigation of the Apollo 13 project [11]. Unlike the original definition in NASA, digital twin technology in transportation engineering now integrates two-way data flow, AI-based dynamic updates [16], and real-time parameter synchronization across diverse applications such as VRU protection [17], IoT cybersecurity [18], risk assessment [19], autonomous vehicle control [20] and so on, thus serving as an important tool for advancing urban mobility and intelligent transportation systems.

One of the differences between digital twin and pure simulation lies within the "Twin" capability of the digital replica for the representation of the digital twin, as well as the dynamic update mechanism for digital twin, which pure simulation lacks [21]. A complete data update process for a digital twin system consists of the following steps: initialization, model construction, and scenario-based prediction. In this process, the digital twin is applied to simulate real-world scenarios, with the collected trajectories used to iteratively refine and update the parameters of the digital twin model. The digital representation of the target entity can run parallel in the digital world, meaning it is possible to utilize the comparatively high computational resources on the cloud side to perform multiple runs under perturbed situations. This can serve as a reference to improve system design [11]. There have been several applications utilizing a similar concept. [21] used the digital replica of the tire rotation model to help adjust the developed controller. In this work, the authors first build a digital replica of the tire model that integrates domain knowledge and the collected data. Then this digital twin of the tire is used to predict the rotation angle of tire under different driving conditions. Finally, the prediction result is utilized by the test vehicle controller. The work presented in [22] also utilizes the digital twin for prediction, and in this work the digital twin was utilized for driver modeling, and the result of the model is integrated into the generation of control commands. There had been some work considering the digital twin with intelligent transportation system, yet not much touched on the part of using the digital twin as a more accurate representation of the evaluation target.

B. Accelerated Testing for HDV/ADAS

Numerous vehicle tests have been performed for the HDV/ADAS system, and among them there is also significant work on accelerated testing, with the aim of having an accurate estimation of the crash rate of vehicles with a significantly lower number of testing cases needed. This kind of problem is called testing scenario library generation (TSLG) [6], which uses a surrogate model (SM) to estimate the crash rate and a Monte Carlo Markov chain (MCMC) to sample the selection of the cases from the generated testing library. Despite the difference in aspect from vehicle engineering and transportation engineering, they all involve the consideration of exposure frequency and criticality of the testing cases themselves. The calculation of the exposure frequency relies on the collected Naturalistic Driving Data(NDD). A key challenge in estimating the exposure frequency from NDD is accurately accounting for events that have not occurred. This issue has been addressed in [23], which focuses on improving estimation methods for such cases.

Accurately creating SM for accelerated testing is also another issue for performance estimation. As stated in [7], the error between the actual vehicle model and the constructed SM is the key contributing factor to the difference between the estimated crash rate and the actual crash rate. For the construction of SM, several works utilized different methods. Research work in computer software engineering [24] directly used the digital representation of middleware and decision systems as SM, as they can be executed multiple times in different environments without directly involving the physical entity of the test vehicle. There may be issues with using only the digital system to evaluate system performance: 1. computation load is large if it is running on all cases 2. no consideration of the physical part of the test vehicle. This means a better representation of VUT is needed. Calibrated car-following models were used by [6], [25]. This proposed static representation of vehicle dynamics replaces the actual testing vehicle for the estimation of the criticality of the crash rate. The issue with calibration of a car-following model is that the newly collected trajectories could not directly contribute to the parameter update, and it takes multiple trajectories for the new car-following model's calibration to converge. Also, there is no selective mechanism of calibrating car-following when new trajectory data are collected. Thus, if the newly collected data differs a lot from the historical dataset, then the calibrated car-following model could not accurately represent the maneuver of the set of trajectories from historical data. Therefore, a representation of VUT with both dynamic update and selective update mechanism is needed, which we represent in our work.

III. Methodology

This work proposes to validate the workflow of online updating digital twin with the corresponding application in the downstream area. The digital twin is built from collected trajectories, and we also include a trajectory acceptance mechanism to make sure the newly included trajectories do not differ from the originally created twin too much. In this section, we will describe the proposed digital twin formulation mechanism, the digital twin model along with the mechanism to determine if the newly collected trajectories will be accepted for parameter update. The entire data update workflow is shown in Figure 2.

A. Digital-Twin Framework and Components

The digital twin representation of our proposed work consists of two parts: the parameter and its corresponding model-

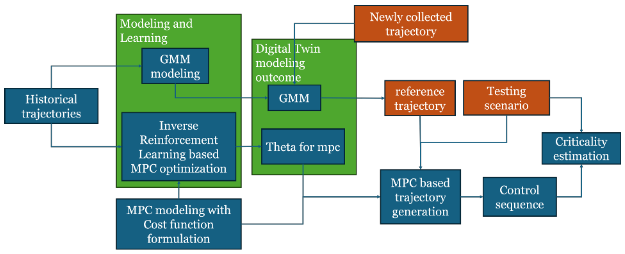

Fig. 2: Data update workflow

predictive control based control generation, and a "reference trajectory" generated from Gaussian Mixture Model(GMM). The GMM was trained from historical collected trajectories and can generate a reference trajectory that has the highest probability of occurrence with weight adjustment given the initial ego vehicle state. The model predictive control-based optimization generates trajectory that are as close to the original collected maneuver as possible in the consideration of safety, mobility, comfort and energy.

Note that our proposed digital twin creation and modeling approach captures the responsive behavior of the vehicle under test. Therefore, both components of the digital twin framework are needed: one for ego vehicle maneuver modeling, the other for responding to surrounding environment. GMM is used to model the maneuver of the ego vehicle and does not consider interference with the surrounding environment. GMM-based reference trajectory generation ensures that, given the initial state of the ego vehicle, the digital twin can generate a vehicle trajectory profile that is close to the collected trajectory cluster. MPC-based control ensures that the vehicle response is close to the historical trajectory cluster.

We can define the whole problem as an optimization problem at the high level. In order to explain the whole problem, we need to introduce some definitions and annotations. First, we define a data point within the trajectory. Each data point on the trajectory is denoted as p_i with definition $p_i = (t_i, s_i, v_i, a_i)$ where t denotes the time frame, s denotes 1D location within the road network, v denotes vehicle longitudinal driving speed, a denotes the vehicle's longitudinal acceleration at that time stamp. Then we denote the whole trajectory for a single record as $T_j = \{p_i = (t_i, s_i, v_i, a_i) | i = 1...N\}$ where N is the fixed trajectory length in our framework, and j is the record id with regard to the collected trajectory set. Note that the original data record for a single vehicle might have multiple trajectory records, as the record for a single vehicle might have a longer time compared to the planning horizon we define in our framework. In that sense, we use rolling horizon to crop the original collected HDV trajectory for multiple sub trajectories, and each sub trajectory is regarded as an independt record in out digital twin system. Then we denote the set of trajectories as $X_k = T_j^k, j = 1...M$ where the upper note of the trajectory record T_j^k refers to the label of the trajectory, and the footnote j refers to the index of trajectory record in the corresponding trajectory set.

The high-level formulation of the digital twin creation problem can be formulated as an optimization problem, with the notation mentioned above. We first define a set of historical trajectories as $H = (X_e, X_l)$ where X_e is the set of ego vehicle trajectories and X_l is the set of corresponding leading vehicle trajectories. Our aim is to build a digital representation that can represent the common maneuvers from X_e while reacting to different background scenarios. The modeling process of the digital twin model itself can be formulated as an optimization problem below:

$$\min_{\pi_k, \mu_k, \Sigma_k, w} J(P^k(X_e), w, X_l) \tag{1}$$

where:

- $P^k(X_e)$ denotes the GMM model trained from the trajectory in set k.
- w: Weight parameter balancing the influence of ego and leading vehicle trajectories.
- $J(P^k(X_e), w, X_l)$: Predefined cost function based on model predictive control formulation to ensure that the learned weights and GMM model can represent historical trajectories of the ego vehicle as close as possible

The GMM model $P^k(X_e)$ is defined as :

$$P^k(X_e) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X_e \mid \mu_k, \Sigma_k) \tag{2}$$

where:

- π_k: Weights of Gaussian components,
- μ_k: Means of Gaussian components,
- Σ_k: Covariance matrices of Gaussian components.

Note that this is the high definition of problem formulation. The modeling of GMM, the formulation of model-predictive control based trajectory generation problem, and trajectory selection for parameter update will be discussed in the following part of this section.

B. Gaussian Mixed Model-based Historical Trajectory Modeling

Certain data augmentation methods have to be used to mitigate the gap between the limited number of collected trajectories versus the required for "ground truth" trajectory when applying for prediction. In our dynamic-updated digital twin for the accelerated evaluation framework, we also introduce a trajectory selection mechanism to ensure the newly collected trajectory could be properly used to update the digital twin model parameter. We introduce a GMM based modeling approach to cope with the two mentioned need.

In our previous work [26], we have cooperated with GMM in the task of modeling crowd-sourced trajectories where GMM shows capability of capturing changing and dynamic patterns in the HDV trajectories in the work zone. In this work, the trajectories are modeled on a 1D control sequence considering time-spatial properties. Therefore, the trajectories are modeled as $T_j = \{(t_i, s_i, v_i, a_i) | i = 1...N\}$ where each i represents the time index within the trajectory starting from the initial state, and the j under the note of T represents the trajectory index. To simplify the problem here, we use fixed-length trajectories and long trajectories are divided as length N with sliding window approach.The modeling of trajectory using GMM is shown as follows:

Trajectory Modeling as a 4D Gaussian Mixture Model (GMM): In order to obtain trajectory modeling with 4D GMM, we need to obtain the probability density function (PDF) of the GMM, which is defined as equation (2). For a given GMM model, Each Multivariate Gaussian distribution is defined through mean, covariance, and the whole GMM model is a combination of multiple Gaussian Models.

Training the GMM: To estimate the parameters π_k, μ_k, Σ_k for each Gaussian component, the Expectation-Maximization (EM) algorithm is used:

E-step: Compute the posterior probabilities (responsibilities) for each data point belonging to each Gaussian component.

$$\gamma_{nk} = \frac{\pi_k \, \mathcal{N}(X_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \, \mathcal{N}(X_n \mid \mu_j, \Sigma_j)} \tag{3}$$

M-step: Update the parameters π_k, μ_k, Σ_k using the responsibilities:

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk} \tag{4}$$

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} X_n}{\sum_{n=1}^{N} \gamma_{nk}} \tag{5}$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_{nk} (X_n - \mu_k)(X_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma_{nk}} \tag{6}$$

GMM Parameters obtained: After training, the GMM provides the following:

- **Component weights (π_k):** Represent the proportion of the data assigned to each Gaussian.
- **Means (μ_k):** Represent the center of each Gaussian component in the 4D trajectory space.
- **Covariances (Σ_k):** Capture the spread and correlations between t, s, v, a within each Gaussian component.

C. Inverse Reinforcement Learning based Control Maneuver Generation

There are several ways to construct a digital twin using human driving data. In this paper, we formulate the trajectory generation problem as an optimal control problem using Model Predictive Control(MPC). The reason for using MPC to formulate the problem is that MPC has been widely used for vehicle control [27]–[29], and they can properly model the decision making process by integrating multiple factors into the decision-making process, similar to human driver decision making process [30]. The weight of MPC is obtained from historical data with inverse reinforcement learning and optimization. Similarly to the method presented in [31], we first construct a nonlinear model predictive control model with cost functions, then we use the inverse reinforcement learning (IRL) method to obtain optimal weights for the cost function with respect to the optimal control model presented.

The trajectory generation model is formulated as a non-linear optimal control model, which we used for the construction of the digital twin model in this paper. The model is represented as follows.

$$\arg\min_{s_{1:N}} \sum_{i=1}^{5} \theta_i f_i(s, u) \tag{7}$$

$$\text{s.t. } u \in U \tag{8}$$

where N is the planning horizon, $s_{1:N}$ denotes the generated control sequence on ego vehicle through out planning horizon considering interaction with surrounding vehicles and similarity to collected trajectories. U is the control set of the target ego vehicle, θ is the weight vector based on the formulation of MPC and is obtained through IRL training. f_i denotes features utilized in the MPC problem, where each feature represents one aspect of the model. Several features are calculated, and a cost function is designed for the optimal control problem. The features used are listed below:

- $f_1 = \frac{1}{N} \sum_1^N a_i^2$, which calculates the average acceleration throughout the planned trajectory.
- $f_2 = \frac{1}{N-1} \sum_1^{N-1} (a_{i+1} - a_i)^2$, which calculates the average acceleration change in each time step throughout the generated trajectory. This feature represents the jerk, and the large jerk is penalized to ensure that a comfortable trajectory is generated.
- $f_3 = \sum_{k=1}^{N} (x_k^n - x_k^i) - \left(s_0 + t_0 \cdot v_k^i + \frac{v_k^i \cdot (v_k^n - v_k^i)}{2ab} \right)$ calculates the safety feature for the ego vehicle during car-following scenarios. This feature tries to model the ego vehicle's response throughout the car-following process in the planning horizon.
- f_4 is the terminal feature that is used to ensure that the generated trajectory also serves as the digital replica of the historical trajectory set.
- f_5 is the stepwise feature that is used to ensure that the generated trajectory can closely represent the ground truth at each step.

As stated, the DT itself consists of modeling for ego vehicle maneuver as well as responsive maneuver. The MPC formulation is used to model the responsive maneuver, and the local optimum for weight factor θ is trained through IRL and serves as part of the DT model.

D. Parameter Update for Digital Twin: More Means Better?

We include a trajectory selection mechanism in our framework in which we have to determine the similarity between the newly collected trajectory and the historical trajectories. The similarity of the trajectory is determined by the probability that the trajectory belongs to the distribution defined by pre-trained GMM, which is shown below:

We have a newly collected trajectory $T_j = \{p_i | i = 1, \ldots, N\}$, where $p_i = (t_i, s_i, v_i, a_i)$. For pretrained digital twin we denote it as $P^k(X_e)$, where the label k represents driving maneuver k and X_e represents the training set of that specific maneuver. $P^k(X_e)$ is described with parameters π_k, μ_k, Σ_k. Here we evaluate the probability that the trajectory T_j belongs to the trajectory collected in set k using the log-likelihood of the trajectory under the GMM, which is calculated as:

$$\log P\big(T_j \mid P^k(X_e)\big) = \sum_{i=1}^{T} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(X_i \mid \mu_k, \Sigma_k) \right), \tag{9}$$

where the multivariate Gaussian probability density function is defined following (2).

If there are multiple m pre-trained GMMs denoted as $P^1(X_e^1)$,...,$P^m(X_e^m)$, then a new digital twin model will be initiated if the new data satisfy none of the pre-trained and historical vehicle maneuvers. The determination of such process involves calculate the log-likelihood of the trajectory under each GMM:

$$\log P(T_j \mid P^k(X_e)), \quad \text{for } k = 1, 2, \ldots, m. \tag{10}$$

The selection of new data will be determined based on the log likelihood, and the new trajectory will be used to update the DT model parameters correspond to the GMM model with the highest log-likelihood:

$$P^*(X_e) = \arg\max_q \log P((T_j) \mid P^q(X_e)). \tag{11}$$

where

$$q \in 1, ..., m$$

And it should also be noted that there exists possibility that new trajectory is not similar with any set of the historical data. In that sense, the inclusion of new trajectory data T_j in the DT model with the label k will only be accepted if and only if $\log P((T_j) \mid P^q(X_e)) \geq \gamma$, where γ is a predetermined threshold.

IV. Case Study

In this part, we present the setup and baseline of the experiment used for comparison.The aim of our case study is to demonstrate that: 1. our proposed new modeling approach for the digital twin model can integrate with the TSLG problem and support the calculation of scenario criticality; 2. the mechanism of selective trajectory integration into the DT model can ensure the representativeness of the DT model created. The content of this section involves the description of simulation environment, the setting for the two different driving scenarios which we used for creation of different DT model. Then we evaluate the created DT model in the evaluation of criticality and then integrate the DT into TSLG problem with NDD data collected from real-life scenarios.

A. Experiment Environment Setup

A microscopic simulation environment using SUMO [15] is used to generate traffic data for both the VUT and the background vehicle for different driving scenarios and working conditions. In the simulation environment, we generate groups of vehicles on a high-definition map, and the geometry of the map is provided by ExiD dataset [32]. The reason for using the HD map from ExiD is that we use the NDD data collected from ExiD to calculate the exposure frequency, and to ensure that the driving maneuver fits the scenario, we use the same working condition and road geometry for the NDD data.

For the same road geometry, we generate two sets of vehicles representing different driving patterns. We use Intelligent Driving Model(IDM) [33] to control background vehicles using the simulation environment provided by SUMO. To ensure that the vehicle maneuver generated is reasonable, we adopt the IDM car-following model parameters from previous published work [34] and implement the two representative car-following maneuvers in SUMO for generation of HDV trajectories, which will be applied in the DT creation and TSLG problem later.

To integrate the DT into the generation of testing library with regard to car-following scenarios, we first model the target scenario and then obtain distribution based on the NDD. To simplify the scenario, we use constant speed assumption for the background vehicle when considering car-following cases. In our scenario, we model the state of the scenario as $r, \dot{r}, v_{ego}$ where r denotes the range between the front vehicle and the ego vehicle, $\dot{r}$ denotes the speed difference between leading vehicle and the ego vehicle, v_{ego} denotes the speed of the VUT. We also limit the range of car-following cases to less than 80 meters, as a higher car-following distance means a lower interaction between the ego vehicle and the leading vehicle and the maneuver of the ego vehicle will be dominated more by the free-flow pattern than by the car-following pattern [35]. To ensure that the generated test library does not overestimate the occurrence of critical events, we calculate the exposure frequency of NDD and have the exposure frequency shown below as:

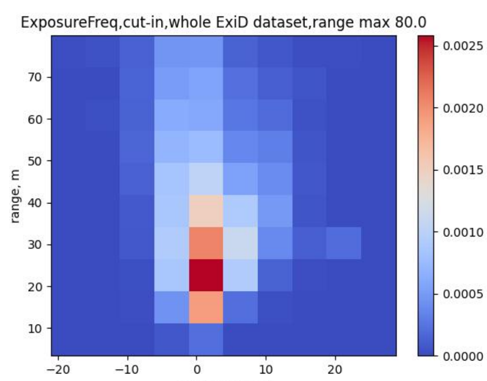

Fig. 3: Exposure frequency for NDD data on test scenario

B. Digital Twin Generation with Certain Set of Trajectories

After collection of HDV trajectories (in this work, we use SUMO-generated trajectories), we perform DT modeling for ego vehicle maneuver and responsive maneuver. For ego vehicle maneuver, we perform GMM modeling based on collected vehicle data. Note that the original modeling is a 4D GMM model, which makes it hard to visualize, and here we just visualize the GMM model for acceleration and speed to illustrate the difference between two kinds of driving maneuvers. It is also notable that the collected data are discrete and there are always the case when historical data do not have relevant record with regard to new initial state of trajectory. In such a sense, the GMM model provides a good estimation of missing data. As shown in Figure 4, the issue of missing data occurs due to the observation is limited, and the GMM provide estimation for such missing data as shown in Figure 5.

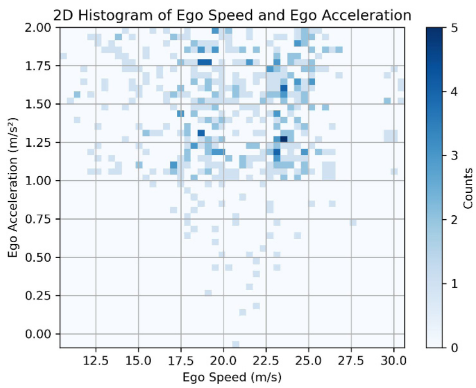

Fig. 4: 2D histogram visualiztion of speed and acceleration from training data

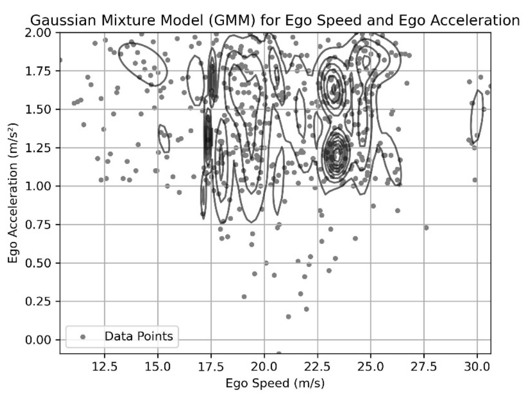

Fig. 5: 2D visualiztion of GMM result based on training data

For modeling of the responsive behavior of the ego vehicle, we follow the formulation of the MPC problem mentioned in equation (8), and we train the parameter of MPC formation based on Inverse Reinforcement Learning shown in [31]. The trained parameter is a local optimum of the historical recorded dataset, and the parameters are obtained only when the training converges, as shown in Figure 6. The trained local optimum parameter at this step, along with the obtained GMM model and predefined MPC formulation in equation (8), forms the DT model as a whole, and we evaluate the capability of "twin" in the following part.

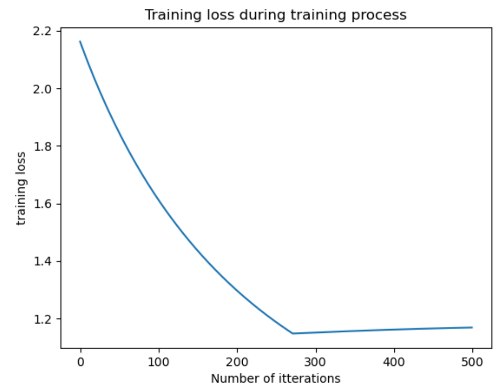

Fig. 6: Convergence figure of IRL training process

C. Evaluation of Similarity for the Proposed Framework

To ensure the correctness of the selected models, we tested the correctness of the DT model based on our simulation method and show the result as follows. To simplify the annotation, we abbreviate ground truth as GT in the following section.

Individual Case illustration: To validate the capability of our proposed model to generate maneuver similar to pre-recorded trajectories while also showing similar maneuver response to background vehicles, we first conducted individual level evaluation and presented the results as follows. At this level, the evaluation focused on comparing the individual twin representations against GT data from test cases. We perform a comparison and evaluation that includes location, speed, and acceleration. Each record was individually analyzed to evaluate the performance of the digital twin model by examining whether the outcome of digital twin model can align with its corresponding ground truth trajectory. We plotted vehicle profile of the ground truth and DT model for visual assessment of the model's performance in replicating historical data. The comparison of outcomes is shown below:

In Figure 7, we plot the time-space diagram for both the ground truth and the result from DT. We compared the location profiles of the vehicle's GT data against the corresponding DT representation. Given that the test scenario focuses on car-following behavior, a time-space diagram was used to visualize and analyze the results. This plot illustrates the spatial and temporal trajectories of both the ground-truth and the DT models, providing a clear comparison of their alignment. From Figure 7 we can see that the location profile generated by DT closely resembles that from ground truth trajectory.

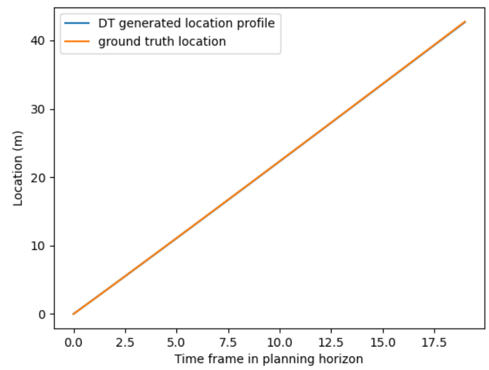

Fig. 7: DT for location comparison

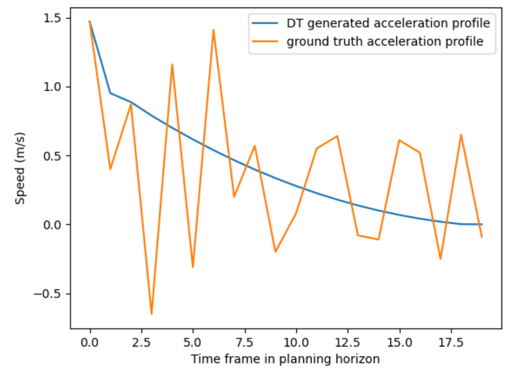

Fig. 9: DT for accceleration comparison

We also plotted the speed profile for evaluation in Figure 8. The divergence in the plot is due to the range of the y axis, which magnifies the difference between DT and ground truth. It can still be seen that the DT's speed profile closely aligns with the GT data. The variations between the two profiles are minimal, indicating DT's capability of resembling the maneuver characteristics of GT. This strong alignment shows the reliability of the DT model in replicating real-world speed behavior.

We also compared the acceleration profile for the vehicle,

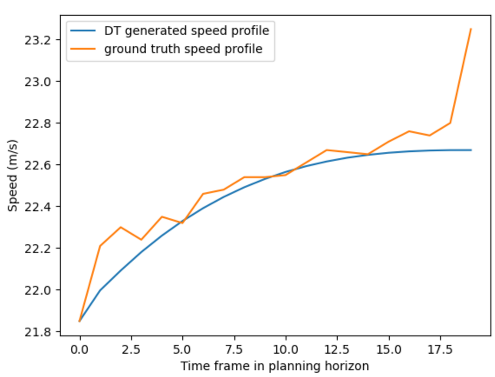

Fig. 8: DT for speed comparison

an example is shown in Figure 9. It is notable that the acceleration profile generated by SUMO shows deviations from expected behavior, as the acceleration profile of the GT data shows a zig-zag maneuver, which actually means a high jerk and uncomfortable driving pattern. However, the DT model effectively mitigates the impact of these deviations, producing a more accurate representation of the vehicle's acceleration dynamics.

Aggregated Result: We then show that our proposed method can obtain a relatively consistent performance in multiple test cases. We show a cumulative result with multiple cases. The testing cases and training cases are separated before the training process began, and the testing cases and training cases are selected under the same kind of driving scenario. This is to ensure the consistency in driving scenario so that the testing result shows the capability of creating digital twin based on selected data:

We aggregated the results from all evaluation cases to provide an overall evaluation of the DT model's performance. We obtain the 2D histogram of Average Displacement Error for all testing cases shown in Figure 10. This shows the ADE for the trajectories generated by DT compared to the corresponding GT in the same scenario record. The histogram reveals that over 80% of the records have an ADE smaller than 0.3 meters, illustrating the model's high accuracy in replicating trajectory profile compared with real-world entities. For location-specific analysis, we further evaluated the final displacement error (FDE) of the trajectories, shown in Figure 11. The corresponding plot shows that the majority of the FDE values are less than 0.6 meters, which is less than the length of a typical vehicle. These findings demonstrate that the DT model consistently produces highly accurate spatial representations, which shows the capability of such model serving as a good surrogate model in downstream applications.

We also evaluated the speed profile by comparing the DT-generated results with the corresponding GT data. A 2D histogram of the ADE for speed demonstrates that over 80% of the records exhibit an ADE smaller than 0.5 m/s. This indicates a strong alignment between the DT and GT speed profiles in most records. The high similarity further supports the reliability of the DT model in capturing speed dynamics effectively throughout the planning horizon.

D. Digital Twin in Test Library Generation

We connect the proposed digital twin modeling mechanism with the TSLG problem to show its application. In this part, we also show the importance of our selective acceptance for new trajectory data. We first show that failing to perform selective

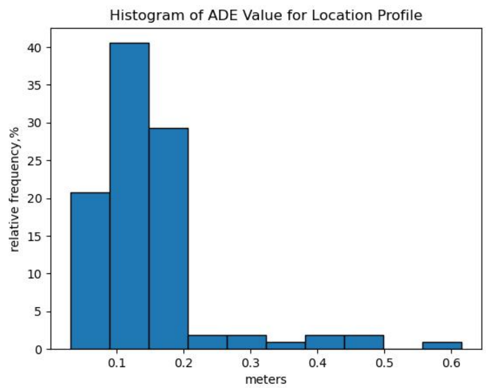

Fig. 10: distribution of ADE for location profile

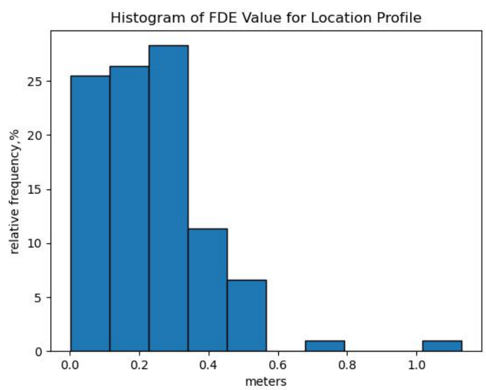

Fig. 11: FDE for location distribution

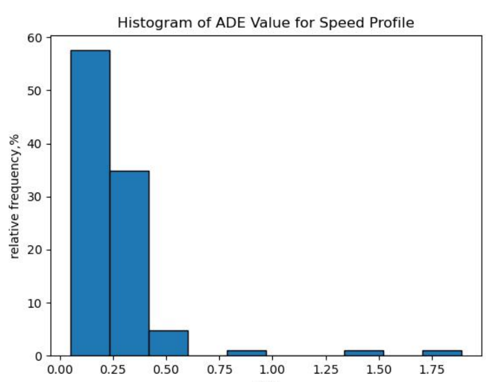

Fig. 12: ADE for velocity distribution

acceptance for new data can downgrade the representativeness of the DT model, then we show the generated testing library of both kinds of maneuvers.

Calculation of Criticality: The case study includes two distinct types of driving maneuvers: cautious and aggressive, implemented based on the findings presented in the research paper [34]. These maneuvers represent different driving behaviors, allowing for a comprehensive analysis of the DT model under varying conditions. Figure 14 presents the results obtained from the calculation of the criticality of the aggressive driving maneuver, highlighting the associated dynamic behavior and potential risks. In contrast, Figure 13 illustrates the criticality results obtained from the cautious driving maneuver, showing a more measured and controlled driving pattern. The comparison of these scenarios provides a baseline for the TLSG problem.

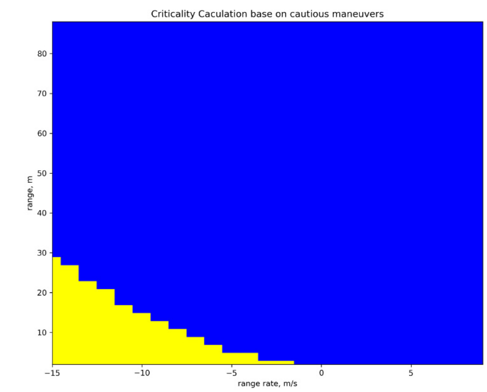

Fig. 13: Criticality calculation result based on collected trajectories for cautious maneuver

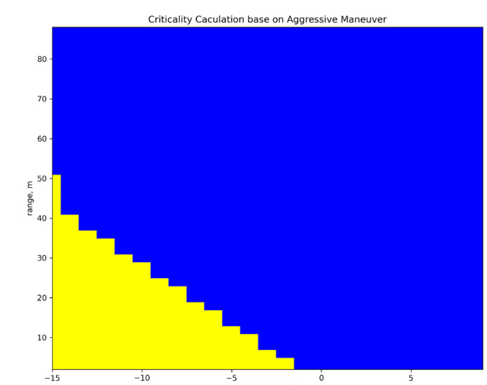

Fig. 14: Criticality calculation result based on collected trajectories for aggressive maneuver

A key feature of our digital-twin model is its selective acceptance mechanism for new trajectories. To illustrate the importance of this module, Figure 15 demonstrates that incorporating new trajectories without proper identification leads to significant errors in the generated DT model and highlights the consequence of including trajectory data that do not belong to the same driving pattern cluster. The lack of identification proposed by our work will downgrade the model's representativeness. This underscores the critical role of trajectory validation in maintaining the accuracy and reliability of the DT model.

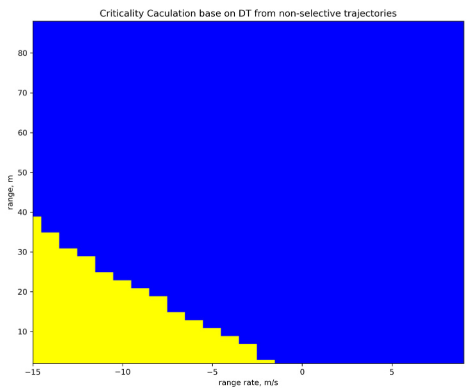

Fig. 15: Criticality calculation result based on collected trajectories without selection mechanism

Testing Library Generation Result: We present the results of the generated test library for both cautious maneuvers (in Figure 17) and aggressive maneuvers (Figure 16). While the criticality differs significantly between the two driving styles, the difference in testing cases diminishes when considering the exposure frequency from NDD. Despite these variations, the proposed DT model effectively captures the underlying behavior of the VUT (VUT). This demonstrates the model's utility as a robust and accurate representation in the testing library generation for vehicle evaluation. As shown in the test library generation result, both generated library considered the criticality and the exposure frequency. The test cases which have high exposure frequency in the NDD dataset also have a higher probability of occurrence in the generated test library. The difference lies within cases that are both highly critical and have some exposure frequency. For scenarios that are dangerous to aggressive behavior but not dangerous to cautious behavior, the testing library generated for cautious behavior does not select those scenarios. The same scenario has been selected for testing library on aggressive maneuver, and this is due to the higher criticality provided by DT.

V. Discussion and Conclusion

This study presents a modeling framework for digital twin models from recorded historical trajectories. We demonstrate

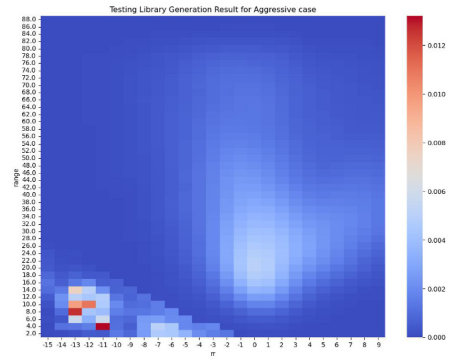

Fig. 16: Result for testing library generated on aggressive maneuvers

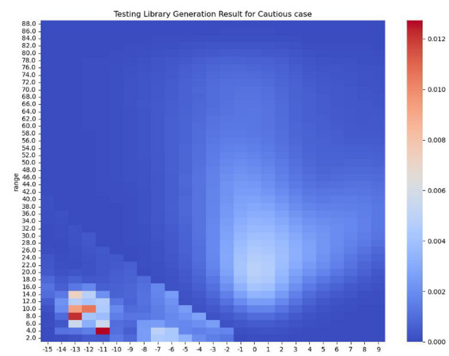

Fig. 17: Result for testing library generated on cautious maneuvers

its capability of generating maneuvers that capture both the ego maneuver of the historical trajectories and how ego vehicles respond to surrounding trajectories. We evaluate the similarity of the DT generated profile with ground truth and show that the proposed trajectory selection framework is important to maintain the representativeness of the DT model.

Although the experiments showcase promising results for downstream applications, there are still some issues that need future explorations, as listed below.

- Reliance on proper modeling for historical trajectories: The digital twin framework can be regarded as a trajectory generation module which also ensures similarity of generated trajectory with new trajectories. In this work, we used GMM to capture the pattern of maneuvers in historical trajectories. This method necessitates the assumption that both the new driving pattern and the working conditions had to be similar. If either one deviates greatly from the historical data, then the capability of DT model to generate similar pattern according to historical data will be downgraded. We plan to address this issue in our future work and release this assumption. We plan to replace the GMM model by introducing predictive models (e.g., LSTM) that can better capture the characteristics in the historical trajectories.

- Ability to distinguish different driving behaviors. As shown in the case study, the difference between different driving patterns can be observed. However, this classification and trajectory selection is based on provided driving pattern labels. Future work needs to develop an automatic driving pattern classification mechanism that relies less on previously collected driving pattern labels.
- extension to spatial-temporal features considering traffic domain knowledge. In this work, all trajectories are cropped using rolling horizon, thus the long-term evolving effect of traffic flow was neglected in such modeling approach. The integration of traffic flow into the modeling process can be considered in future work.

References

[1] Xinpeng Wang, Songan Zhang, and Huei Peng. Comprehensive safety evaluation of highly automated vehicles at the roundabout scenario. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):20873–20888, 2022.

[2] Xinpeng Wang, Tinghan Wang, Shaobing Xu, Yuanxin Zhong, and Huei Peng. The implementation of safety acceptance evaluation for highly automated vehicles. *International Journal of Automotive Engineering*, 14(3):58–65, 2023.

[3] Hasan H. Demirarslan, Y. Chan, and M. Vidulich. Visual information processing: Perception, decision, response triplet. *Transportation Research Record*, 1631:35–42, 1998.

[4] Heejin Jeong and Yili Liu. Modeling of stimulus-response secondary tasks with different modalities while driving in a computational cognitive architecture. *Proceedings of the Driving Assessment Conference*, 9:58–64, 2017.

[5] Songan Zhang, Huei Peng, Ding Zhao, and H Eric Tseng. Accelerated evaluation of autonomous vehicles in the lane change scenario based on subset simulation technique. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3935–3940. IEEE, 2018.

[6] Shuo Feng, Yiheng Feng, Chunhui Yu, Yi Zhang, and Henry X Liu. Testing scenario library generation for connected and automated vehicles, part i: Methodology. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1573–1582, 2020.

[7] Shuo Feng, Yiheng Feng, Haowei Sun, Shan Bao, Yi Zhang, and Henry X Liu. Testing scenario library generation for connected and automated vehicles, part ii: Case studies. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):5635–5647, 2020.

[8] Xianan Huang, Songan Zhang, and H. Peng. Developing robot driver etiquette based on naturalistic human driving behavior. *IEEE Transactions on Intelligent Transportation Systems*, 21:1393–1403, 2018.

[9] Stefan Boschert and Roland Rosen. Digital twin—the simulation aspect. *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers*, pages 59–74, 2016.

[10] NASA. Digital twins and living models at nasa. Technical report, NASA Technical Reports Server (NTRS), 2021. Accessed: 2025-01-09.

[11] Siemens Digital Industries Blog. Apollo 13: The first digital twin. https://blogs.sw.siemens.com/simcenter/apollo-13-the-first-digital-twin/, 2021. Accessed: 2025-01-09.

[12] Ruikun Luo and Dmitry Berenson. A framework for unsupervised online human reaching motion recognition and early prediction. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2426–2433. IEEE, 2015.

[13] A. Ehlers, R. Mayou, and B. Bryant. Psychological predictors of chronic posttraumatic stress disorder after motor vehicle accidents. *Journal of abnormal psychology*, 107 3:508–19, 1998.

[14] Ronald C. Kessler, Katie A. McLaughlin, Melany A. Green, Hans-Ulrich Wittchen, and Jordi Alonso. Trauma and ptsd in the who world mental health surveys. *European Journal of Psychotraumatology*, 8(sup5):1353383, 2017.

[15] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. IEEE, 2018.

[16] United States Department of Transportation (DOT). Emergency traffic management digital twin: Prototype and pilot implementation. Technical report, USDOT, 2023. Models city traffic with real-time feeds for evacuations, ambulance routing, multi-level 3D environment.

[17] Federal Highway Administration (FHWA). Research on vulnerable road user (vru) proactive safety using digital twins. Technical report, United States Department of Transportation (DOT), Federal Highway Administration, 2022. Focus on real-time data ingestion, collision prediction with digital twin shadow/sibling framework.

[18] Philip Empl and Günther Pernul. Digital-twin-based security analytics for the internet of things. *Information*, 14(2):95, 2023.

[19] Enrico Zio and Leonardo Miqueles. Digital twins in safety analysis, risk assessment and emergency management. *Reliability Engineering & System Safety*, page 110040, 2024.

[20] Ziran Wang, Chen Lv, and Fei-Yue Wang. A new era of intelligent vehicles and intelligent transportation systems: Digital twins and parallel intelligence. *IEEE Transactions on Intelligent Vehicles*, 8(4):2619–2627, 2023.

[21] Jianlong Wang, Chuanwei Zhang, Zhi Yang, Meng Dang, Peng Gao, and Yansong Feng. Research on digital twin vehicle stability monitoring system based on side slip angle. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[22] Xishun Liao, Ziran Wang, Xuanpeng Zhao, Kyungtae Han, Prashant Tiwari, Matthew J Barth, and Guoyuan Wu. Cooperative ramp merging design and field implementation: A digital twin approach based on vehicle-to-cloud communication. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):4490–4500, 2021.

[23] Xintao Yan, Zhengxia Zou, Shuo Feng, Haojie Zhu, Haowei Sun, and Henry X Liu. Learning naturalistic driving environment with statistical realism. *Nature communications*, 14(1):2037, 2023.

[24] Chejian Xu, Wenhao Ding, Weijie Lyu, Zuxin Liu, Shuai Wang, Yihan He, Hanjiang Hu, Ding Zhao, and Bo Li. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[25] Max Winkelmann, Constantin Vasconi, and Steffen Müller. Transfer importance sampling–how testing automated vehicles in multiple test setups helps with the bias-variance tradeoff. *arXiv preprint arXiv:2204.07619*, 2022.

[26] Hanlin Chen, Renyuan Luo, and Yiheng Feng. Improving autonomous vehicle mapping and navigation in work zones using crowdsourcing vehicle trajectories. *arXiv preprint arXiv:2301.09194*, 2023.

[27] Anye Zhou, Zejiang Wang, and Adian Cook. Model predictive control-based trajectory shaper for safe and efficient adaptive cruise control. In *2023 IEEE International Automated Vehicle Validation Conference (IAVVC)*, pages 1–7, 2023.

[28] Hao Zhou, Anye Zhou, Tienan Li, Danjue Chen, Srinivas Peeta, and Jorge Laval. Congestion-mitigating mpc design for adaptive cruise control based on newell's car following model: History outperforms prediction. *Transportation Research Part C: Emerging Technologies*, 142:103801, 2022.

[29] Siddharth H Nair, Eric H Tseng, and Francesco Borrelli. Collision avoidance for dynamic obstacles with uncertain predictions using model predictive control. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 5267–5272. IEEE, 2022.

[30] Ardalan Vahidi, Anna Stefanopoulou, and Huei Peng. Adaptive model predictive control for co-ordination of compression and friction brakes in heavy duty vehicles. *International Journal of Adaptive Control and Signal Processing*, 20(10):581–598, 2006.

[31] Jun Ying and Yiheng Feng. Full vehicle trajectory planning model for urban traffic control based on imitation learning. *Transportation research record*, 2676(7):186–198, 2022.

[32] Tobias Moers, Lennart Vater, Robert Krajewski, Julian Bock, Adrian Zlocki, and Lutz Eckstein. The exid dataset: A real-world trajectory dataset of highly interactive highway scenarios in germany. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 958–964, 2022.

[33] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, 2000.

[34] Yongyang Liu, Anye Zhou, Yu Wang, and Srinivas Peeta. Proactive longitudinal control to preclude disruptive lane changes of human-driven vehicles in mixed-flow traffic. *Control Engineering Practice*, 136:105522, 2023.

[35] P.G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105–111, 1981.