DRIVER BEHAVIOR MODELING

by

Ferhat Melih Dal

B.S., Electrical & Electronics Engineering, Boğaziçi University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2022

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Lale Akarun for her guidance and support and for bearing with the ups and downs that I went through during my MS journey.

I am grateful to Assist. Prof. Berk Gökberk and Assist. Prof. Ebru Arısoy-Saraçlar for taking part in my thesis jury. Their valuable comments have led me in a path that I am glad to have walked.

My family is like a backbone that supports me throughout my life and writing this thesis has not been an exception. I am forever grateful to my mother Ayşe, my father Ali, my sister Kubra and my brother Yunus.

I want to thank my friends who have shown great support throughout this journey. With their support, I did not feel alone in my struggles, Tuna Han Salih Meral, Huseyin Temiz, and Korhan Polat.

Lastly, I want to thank Computer Science Department at Bogazici. The moments I spent with my dear professors in this time period will be a memory that I will cherish my whole life.

ABSTRACT

DRIVER BEHAVIOR MODELING

Autonomous vehicles are set to be a part of everyday traffic. Their presence in traffic dominated by humans possesses some challenges. Any experience with driving in traffic shows us that each driver is unique in their driving style. So far, this richness in differences in human behavior has not been projected into the models used in traffic simulations. These models are an essential part of the development of autonomous vehicles; from the inference of other vehicle intentions to virtual testing. Therefore creating a more realistic traffic environment is a very important task. In this work, a deep dive into the state of the problem is given. Then, a framework that accounts for different driving styles, as well as different vehicle types, is introduced. Firstly, an in-depth analysis of distinct patterns of driving is carried out in the dataset. Then these distinct patterns are modeled with simulated agents using reinforcement learning. In inference time, a traffic scene is observed, each vehicle is assigned to the pre-trained driver model and a simulation is carried out. As a result, a traffic scene is reconstructed with data-validated models. This new approach that incorporates previous driver modeling work with a behavioral component, paves the way for a more realistic model of the traffic. This realistic traffic model can be used in AV testing and validation.

ÖZET

SÜRÜCÜ DAVRANIŞI MODELLEME

Sürücüsüz araçlar trafiğin önemli bir parçası olma yolunda ilerliyorlar. İnsanların çoğunlukta olduğu bir trafikte sürücüsüz araçların kullanılması bazı zorluklar getiriyor. Trafikte araç sürdüğümüz her hangi bir deneyim bize gösteriyor ki, her sürücü kendi sürüş stilinde farklıdır. Bu sürüş zenginliği trafik simülasyonlarındaki modellere henüz aktarılmamıştır. Bu trafik modelleri, testlerden anlık davranış tahminine, sürücüsüz araç çalışmasının önemli bir noktasıdır. Dolayısıyla, trafiğin gerçekçi bir modelini oluşturmak çok önemli bir problemdir. Bu çalışmada, problemin kökenine inen bir analiz mevcuttur. Sonra, farklı sürüş metodlarını kapyasan bir model önerilmiştir. Öncelikle, veri seri üzerinde derin bir sürücü davranış modeli çalışması yapılarak belirgin davranış örüntüleri belirlenmiştir. Sonra bu belirgin özellikler pekiştirmeli öğrenme kullanılarak modellenmiştir. Değerlendirme zamanında, bir trafik sahnesi gözlemlenmiş, her bir araç önceden modellenen araçlarla eşleştirilmiştir. Sonuç olarak bir trafik sahnesi veri ile doğrulanmış modeller kullanilarak yeniden yaratılmıştır. Araç davranışını simülasyona entegre eden bu yeni yaklaşım trafiğin daha gerçekçi modellenmesi için bir adım oluşturmaktadır. Bu gerçekçi trafik modeli sürücüsüz araç test ve validasyonu için önemli bir adım oluşturmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a^t	Action at time t
b_i^t	Internal state
n	Observation length
p	Prediction length
s^t	Environment state at time t
u_i^t	Control action
w_i	Weights of reward terms
x_i^t	Physical state
z_i^t	Observation
$Q(s,a)$	Q function
$V(s)$	Value Function
π	Policy

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AV	Autonomous Vehicle
DBM	Driver Behavior Modeling
DQN	Deep Q-Learning
HighD	Highway Driving Dataset
IDM	Intelligent Driver Model
MLP	Multi Layer Perceptron
NHTSA	National Highway Traffic Safety Administration
NN	Neural Network
POSG	Partially Observable Stochastic Game
RL	Reinforcement Learning
SOM	Self Organizing Map
SOTA	State of the Art
TC	Trajectory Clustering
TP	Trajectory Prediction

1. INTRODUCTION

1.1. Motivation & Background

Driver behavior modeling is an important problem for a range of topics including, autonomous vehicles(AV), urban planning, and traffic safety research. Subtasks of these topics are; motion prediction, virtual testing & verification, emissions calculations, crash analysis, and prevention. In each of these applications, the aim is to capture how a driver of a vehicle would act in certain situations. Therefore the problem inherently includes the modeling of a vehicle, the interaction of several vehicles, and their interaction with the infrastructure. In the scope of this work, the focus is on *simulation, virtual testing*, and *motion prediction* for AVs.

In a study of 723 accidents involving 1243 drivers, NHTSA concluded that 90% of all the accidents were due to driver error [1]. Another study suggests that there are 450.000 accidents [2] of lane change and merge in the US every year. It's easy to think that AVs will automatically solve this problem, however, in a recent study of AV accidents, it was shown that 86% of all the accidents that the AVs [3] made were because of other, human-driven vehicles rear-ending and side-swiping. Therefore the challenge to model human behavior into testing and verification frameworks of AVs is of utmost importance.

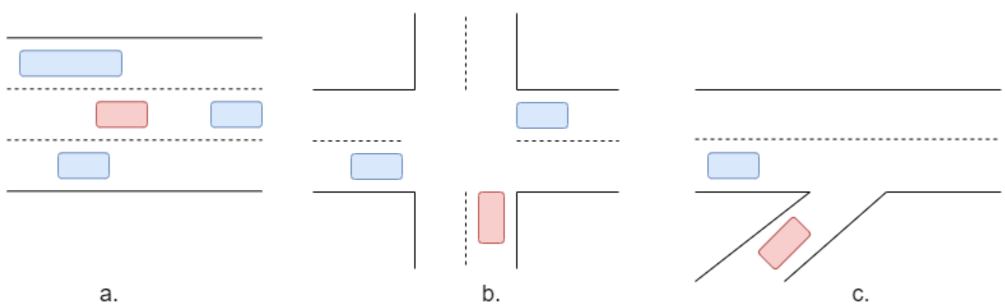

Figure 1.1. Different scenarios of driver behavior problem. a) Highway. b) Uncontrolled Intersection. c) Merging.

At its core, driver behavior modeling can be defined as an "n-agent discrete-time partially observable stochastic game" [4], [5]. In a general traffic scene, the driver is faced with other agents, has inputs from surroundings, an internal state of self, and takes a control action with the imperfect observation that is in accordance with her world model and internal values. The variables in a partially observable stochastic game, POSG, is defined in table 1.1. The elements of interest change according to the application domain. In the motion prediction domain, the aim is to find x_i^t, namely the physical states of each vehicle. In contrast, in the intent estimation task, the goal is to model b_i^t, the internal states of each driver.

Table 1.1. Set of variables in a general POSG formulation as in [5]. A problem of driver behavior modeling can have several or all of the elements in the formulation. Subscripts for agents, superscripts for time.

POSG Elements	
$x_i^t \in X_i$	physical state
$b_i^t \in B_i$	internal state
$u_i^t \in U_i$	control action
$z_i^t \in Z_i$	observation
$z_i^t \backsim G_i(x_1^t, ..., x_n^t)$	observation function
$b_i^{t+1} \backsim H_i(b_i^t, z_i^t)$	internal state update function
$u_i^t \backsim \pi_i(b_i^t)$	policy function
$x_i^{t+1} \backsim F_i(x_i^t, u_i^t)$	state transition function

Depending on the target domain, there are several approaches to the problem of driver behavior modeling. Firstly, the problem can be formulated as a trajectory prediction problem: *given some observed trajectory with* $(x_i^{t-p}, ..., x_i^t)$, *what is the most likely next n points in the space that the vehicle will be in,* $(x_i^{t+1}, ..., x_i^{t+n})$ The above type is generally used where there is a need for online estimation of the vehicle's behavior when interacting with its environment.

Another set of approaches come from the need for correctly forecasting a traffic flow. This type of formulation is more interested in the aggregate behavior than a singular

driver behavior. Here the problem is formulated, *given a set of road structures and a certain distribution of vehicles on these roads, what will be the distribution in some t time later in the future.*
The third main category of the formulation is more related to assessing a driver's unique properties some of which can be intrinsic, such as *aggressiveness*, and some can be related to the current state of the driver, such as *weather*, *road conditions*. And the formulation is *given intrinsic and extrinsic parameters, how will these parameters change the decisions that this particular driver gives differently than the average driver would give?* The fourth and final category of approaches is the mathematical modeling of driving behavior to use in simulators. The formulation of this problem is, *given an initial condition and a goal position, what is the next acceleration and steering to get there?*

An important missing piece in the literature is adding real-data validation to simulation-based methods. Simulation-based methods offer a high degree of freedom in both vehicle and driver behavior selection. However, in many of highly cited traffic simulation papers, [6], [7], there are no strong links between real driving datasets and simulation. This work offers to address this problem by matching the simulated driver behaviors with data extracted behavioral patterns. The basic idea is clustering trajectories of vehicles in behavioral patterns and then matching them to simulated behavioral models. Then this model is validated using a real-world highway driving dataset, HighD, [8] and compared against different baseline methods.

1.2. Contributions

This work brings together several of the SOTA methods in reconstructing a highway scene in a simulation and offers an improvement in reconstruction. It is one of the first works that bring real data validation to simulation models. The discussion of different behavioral patterns in highway driving discussed in this work is another point of contribution. This way future research can benefit from understanding the behavioral complexity posed by different driver behavior. Finally, all code used in this work will be published as open-source code. This will be an important contribution as the link between real traffic data and simulation is missing in the current open-source AV research environment. For all these reasons, the title of this thesis is kept general, because along with proposing a new method, it provides several utilities for researchers and engineers working in the AV testing & verification domain.

Summary of contributions are:

(i) Implementing several SOTA methods.
(ii) Using HighD [8] dataset, evaluate these methods in the same environment.
(iii) Presenting a novel way of unsupervised behavior matching between simulated models and real traffic observation.
(iv) Reconstruct real traffic data in a simulation environment.
(v) Deep analysis into behavioral differences in traffic.
(vi) Open-source traffic data to simulation code to aid AV research.

1.3. Organization of the Thesis

The organization of the rest of this theses is as follows: In Chapter 2 a background into traffic simulation and trajectory prediction models is given. The modeling and training procedures are described in detail. In Chapter 3, details into the proposed method of unsupervised behavior matching is provided. Experiments are presented in Chapter 4with a discussion into intrinsic behavioral patterns in the data. Finally, in Chapter 5, a discussion into the next steps into the driver behavior modeling research is provided.

2. BACKGROUND

Driver behavior modeling (DBM) is an area of work that interests many different disciplines from traffic safety, and psychology to AV research. There have been 3 notable surveys in the field. In [9] the task is formulated as a psychological and traffic aggression problem. Lefevre et.al, [10] approach the problem as a motion prediction problem. The most recent for comes from Brown et.al. [4], which creates a taxonomy for the problem of driver behavior modeling. In the scope of this work, the modeling tasks related to AV testing and simulation have been investigated thoroughly.

Table 2.1. Comparison of related methods in the literature.

	Long Term Planning	Vehicle Type Variation	Behavior Variation	Rich Maneuvers	Real Data Validation	Adjustable Properties
MP	No	No	Yes	Yes	Yes	No
BP	Yes	No	Yes	Yes	Yes	No
MB	No	Yes	Yes	No	Yes	Yes
TS	Yes	Yes	Yes	Yes	No	Yes
TS+DV	Yes	Yes	Yes	Yes	Yes	Yes

Subtasks of DBM in the context of AV research can be presented in 6 main categories:

(i) *Motion Prediction (MP)*, is the task of predicting future trajectory of vehicles, $(x_i^{t-p}, ..., x_i^t)$, after observing their past trajectories, $(x_i^{t+1}, ..., x_i^{t+n})$.

(ii) *Intention Estimation (IE)*, is predicting the action of vehicles at a higher level. Examples of this would be; *stop, turn-right, lane-change, and take-over.*

(iii) *Trait Estimation (TE)* focuses on labeling the trait of drivers usually according to their cooperativeness. It can range from a driver's driving skills to addressing distraction and fatigue levels.

(iv) *Traffic Simulation (TS)*, is simulating an arbitrary traffic scene in a simulator with different vehicle types and driver behaviors. The use cases in BP range from urban planning, and emissions predictions to AV testing.

(v) *Behavior Planning (BP)* addresses the situation where an agent, maybe AV, is planning its actions, such as changing lane, or merging, in a given traffic scenario.

(vi) *Model-Based (MB)* is the name given to the mathematical models of human driving that have been the foundation of TS and BP. Although effective, these models remain very simple for the current level of research in AVs.

The final category of work that is a missing piece in the literate is the part that this work is intending to address; *Traffic Simulation with Real Data Validation, (BP+DV)*. This category does not yet exist in the literature but is needed to capture realistic simulations.

2.1. Traffic Simulation

Simulation of traffic is used from city modeling, emissions measurement, civil engineering, autonomous driving training, and testing. In the scope of this work, only simulations related to AVs will be discussed.

Since traffic needs vehicle models, and their drivers, driver behavior modeling has long been part of this area of work. One of the benchmark works has been the development of the Simulation of Urban Mobility(SUMO) [11] simulator. There are many vehicles, and emissions types to choose from. The driver behavior on this simulator has been the Intelligent Driver Model(IDM) [12] which is a mathematically defined model of the human driver based on reasonable assumptions of rational actors. Given the parameters in Figure2.2.

Traffic simulations are studied under two main categories; macroscopic and microscopic simulations On a macro scale, the aim can be finding congestion and addressing them from simulations, [13]. One other use case for macro simulations is urban planning. Road networks and driver models are used to understand the bottlenecks and plan ahead, [14]. An image from the SUMO simulator is provided in figure 2.1 . This simulator can simulate entire road networks and can be used to model traffic or emission modeling.

Figure 2.1. A snapshot from a popular SUMO [11] environment. A city road network is simulated.

Table 2.2. Intelligent Driver Model parameters for a general vehicle.

Variable	Description	Value
v_0	Desired Velocity	$30m/s$
T	Safe Time Headway	$1.5s$
a	Maximum Acceleration	$0.73m/s^2$
b	Comfortable Deceleration	$1.67m/s^2$
δ	Acceleration Exponent	4
s_0	Minimum Distance	$2m$
–	Vehicle Length	$5m$

Microscopic simulations of traffic are very important for AV testing, training, validation, and verification. These systems are generally created with simplification, to test/train a component of the AV mechanism. In [15], a micro highway simulation environment is created to test tactical decision-making. Oyler et.al. create a highway simulation environment for vehicle testing [16] which is based on the game-theoretic formulation of vehicle interaction. Yoo et. al. [17], use a game theoretical approach for lane-merging and demonstrate the idea with a traffic simulation. Hu et.al. [7] approach the same problem of lane-merging by considering a wider range of road networks.

2.1.1. Agent Models

In deep learning traffic simulations, agents are NNs that take an observation of the environment and output an action. Taking [6] as a reference work, the environment in which the agent is modeled can be seen in the figure. Here the observation space is the relative location and relative velocity of all the vehicles around each vehicle, in a vector representation. The action space is a discrete control space is:

- Hard acceleration.
- Soft acceleration.
- Hard deceleration.
- Soft deceleration.
- Constant speed.
- Turn right.
- Turn left.

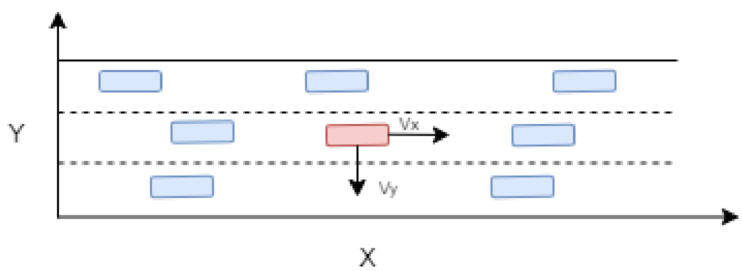

Figure 2.2. Traffic simulation environment in [6].

The observation space varies across works and simulation environments. Following on the work of [6], an example traffic scene is depicted in figure 2.2. From this environment, an example observation space can be defined as:

- Same lane following vehicle X-velocity difference,
- Same lane following vehicle x-location difference,
- Right lane preceding vehicle x-velocity difference,
- Right lane preceding vehicle x-location difference.

Here the observation space is a vector of all the information about the surrounding vehicles, the action space is a discrete value. A typical MLP of 5 layers deep is designed to infer a vector of observations into a discrete action. The training of this network is discussed in the next section.

2.1.2. Deep Q-Learning

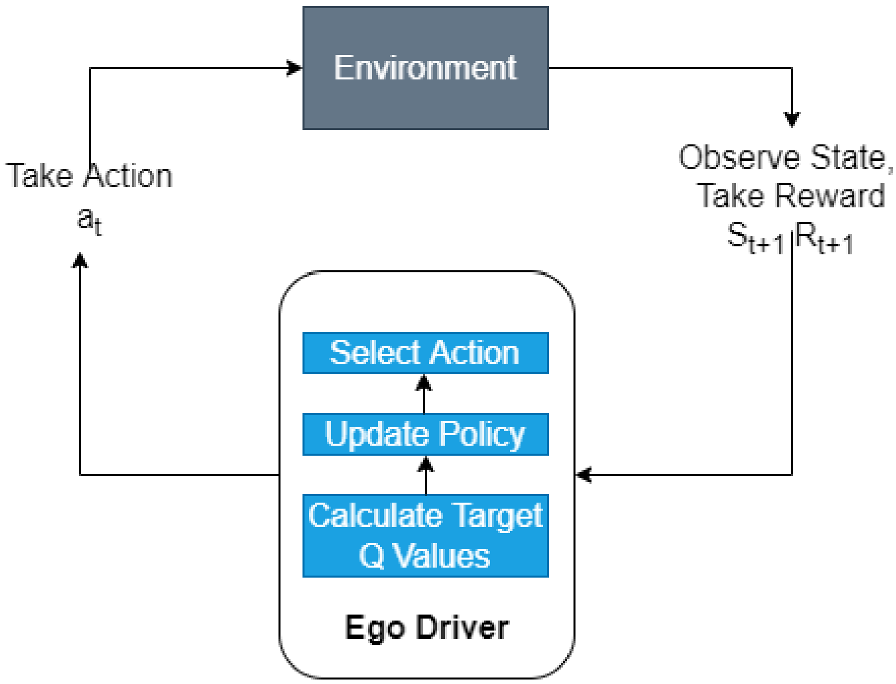

Figure 2.3. Deep Q-Learning training scheme. Agents are trained using an environment which gives responses against actions. Observation from the environment, together with the reward output is taken into account by the agent to update policy. This way, agent learns to take the action that takes the maximum reward from the environment.

Deep Q-Learning is a method of Reinforcement Learning [18] which has been shown to adapt to many different problem with the same training scheme [19]. RL is a great method of semi-supervised learning. By providing observations and their corresponding rewards after each action, differentiable learning can be formulated. The basic idea behind Q-Learning is keeping a table of all the states and their values in a table. Then updating that table of values as the agent moves along the environment. In the case of highway traffic, a state $s_y \in S$ is the state of an agent at time t with the observation discussed above. An action, $a_t \in A$ is a set of actions taken at time t. A policy, is the mapping from states to actions, $\pi : S \longrightarrow A$. An action-value function, $Q(s,a)$ estimates the cumulative reward starting from an action, state pair. It is a measure of how valuable each action is at given state.

The basics components of Deep Q-Learning are *value function*, *policy*, *Q-function* and the update equation for the Q-function. The term *deep* comes from the fact that these components are approximated by deep neural networks The value function of an agent, *cumulative discounted reward starting from current state s*,

$$V^{\pi}(s) = E(\sum_{t>0} \gamma^t r_t), \tag{2.1}$$

where r is the reward and γ is the discount factor. The optimal value function is,

$$V^*(s) = \max_{\pi} V^{\pi}(s). \tag{2.2}$$

The optimal Q-function and optimal value function are related as,

$$V^*(s) = \max_{a} Q^*(s,a). \tag{2.3}$$

Furthermore, optimal policy, defined in terms of optimal Q-function is,

$$\pi^*(s) = arg \max_{a} Q^*(s,a). \tag{2.4}$$

Bellman equation provides a relationship between the value of a state-action pair (s,a) and its successor pairs with,

$$Q(s,a) = E[r_{t+1} + \gamma \max_{\hat{a}} Q(s_{t+1}, \hat{a}) | s_t = s, a_t = a]. \tag{2.5}$$

Finally, based on the Bellman equation, the update to Q-function is iteratively,

$$Q_{t+1}(s_t, a_t) = Q_{t+1}(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a} Q_t(s_{t+1}, a_t) - Q_t(s_t, a_t)), \tag{2.6}$$

where α is the learning rate. In Deep Q-Learning, the Q function is approximated with MLP. Therefore, for a given state, and action, the future cumulative reward is approximated. In its basic form, action is taken by choosing the action which leads to the highest reward approximated by the MLP.

2.2. Vehicle Trajectory Clustering

Clustering is a useful method to make sense of data in an unsupervised manner. The basic idea is to use some metric to separate the data into distinct groups. Trajectory clustering tries to apply these methods to find different sets of trajectories corresponding to a specific property in the data. In this work, specifically vehicle trajectories has been studied.

At table 2.3, the studied papers have been listed with their corresponding features with regard to their approaches, data, and metrics. *Clustering Type* refers to which clustering method is borrowed from the literature such as; agglomerative, kMeans, and spectral. *Encoder* denotes the name of the method used to encode the trajectories if there is an encoder used. *Distance Metric* column is used to denote which metric is used to measure the distance between trajectories so that clustering methods can be applied. *Dataset* denotes which kind of data is used; GPS recordings, in-vehicle recordings, synthetic. Lastly, all papers evaluate their clustering success in some way, and that evaluation metric is denoted by *the Evaluation Criterion.*

Siami et.al. [20], show that resultant clusters can be semantically labeled into meaningful trajectories such as; *warm stopping*, *driving at normal speed* etc. Making semantic meaning of clusters does not affect any evaluation, but it gives a sign that the studied trajectories are valid. In [21], Wang et. al. create their own dataset using a game engine and use these trajectories to find clusters. These clusters are then used to find matching neural networks that imitate these clusters.

Current methods capture single-vehicle trajectories in a meaningful way in their application domain. When these methods are applied to highway driving, they fail to make clear distinctions. None of the current methods account for interaction of vehicles with one another. Which is an essential part of vehicle trajectories in any setting, especially in highway driving.

Table 2.3. Vehicle trajectory clustering survey.

Paper	**Clustering Type**	**Encoder**	**Distance Metric**	**Dataset**	**Evaluation Criterion**
[22]	Agglomerative	-	DTW, LCSS	GPS	Beetween-like
[23]		Seq2Seq		GPS, Synthetic	Classification
[24]	Agglomerative, Spectral	-	Hausdorff, DTW, LCSS	GPS	Classification
[25]	DBSCAN	Auto-enc	-	GPS	Heuristic
[26]	FastICA, PCA	Auto-enc	DTW, LCSS	GPS	Linear seperability
[20]	kMeans, Spectral	Auto-enc	DTW, LCSS	GPS	Davies, Calinski
[21]	Agglomerative	-	Euclidian	Vehicle Recordings	-

2.3. Datasets

There are several highly used datasets in this field of research however, it is worth noting that there is still no publicly available driving dataset that focuses on differences in driving behavior. What the field has done, is to take motion prediction-related datasets and model them as a tool for driver behavior. The first major dataset is the 2007 NGSIM dataset [27], published by the US Federal Highway Administration. It includes several highway recordings, captured by CCTV cameras and processed. The updated version of this can be considered as HighD dataset, [8]. It is 100 hours of trajectory recordings of German highways with high-definition drone cameras. An excerpt from the dataset is provided in the figure 2.4. These two datasets are the major highway datasets. An interaction dataset is also available that focuses on merge, exit, and roundabout scenarios, [28].

Urban driving datasets have been published by leading AV companies in recent years. The first of these urban driving datasets is Argoverse, [29]. Lyft, another AV company published an urban driving dataset, along with a path prediction challenge in 2020, [30]. Lyft dataset is 1001 hours of length of driving provided as trajectories and structures around them in San Francisco. The most recent dataset is published in November 2021 by Waymo, a child company of Google, [31].

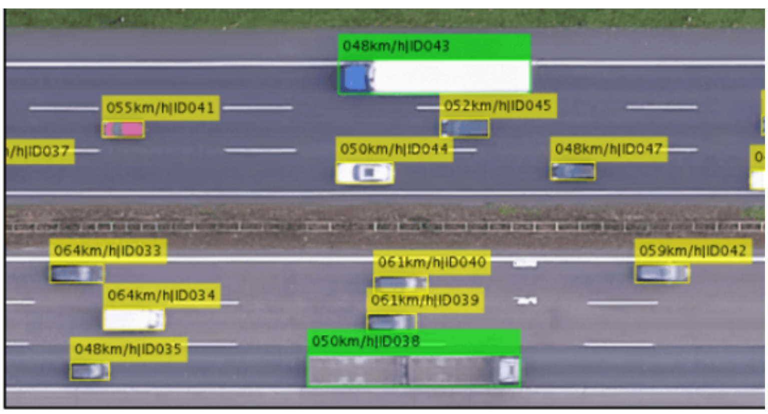

Figure 2.4. An example scene from HighD dataset.

3. UNSUPERVISED BEHAVIOR EXTRACTION & MATCHING

The aim of unsupervised behavior matching is to infer the behavior of a vehicle by observing its trajectory for a short time, then inferring about its motion in the future. A clustering method is used to cluster each trajectory into finite set of behaviors, using the trajectory at *observation time* length, then matching the observed pattern with a simulated behavioral model in an unsupervised manner. This way, a traffic scene can be reconstructed from observations, and this enables a variety of applications including trajectory prediction, testing to verification.

Inspiration for the unsupervised behavior matching method comes from merging two ideas from the literature on driver modeling, traffic simulation [6] and trajectory prediction [32], [33]. In traffic simulation, it is possible to model different vehicle types and different driver behaviors but currently, the method to use these models in reconstructing a real traffic scene does not exist. In contrast, the trajectory prediction literature works with real traffic data but lacks rich maneuvers and a long-term approach. By clustering the data into segments of behavioral patterns, then matching them with simulated models into a way of reconstructing a traffic scenario is possible.

Figure 3.1 shows the outline of our approach. A trajectory clustering is performed from the dataset to identify groups of trajectories. These trajectories ideally represent distinct behaviors in certain time frames. In [20], the number of different behavioral patterns found from trajectories is 27. However, the dataset in [20] consists of city traffic with traffic lights, many turns and merges. Therefore, the relevant trajectory number is a much smaller number in HighD dataset. The relevant time horizon for a behavioral pattern is taken to be 5 seconds.

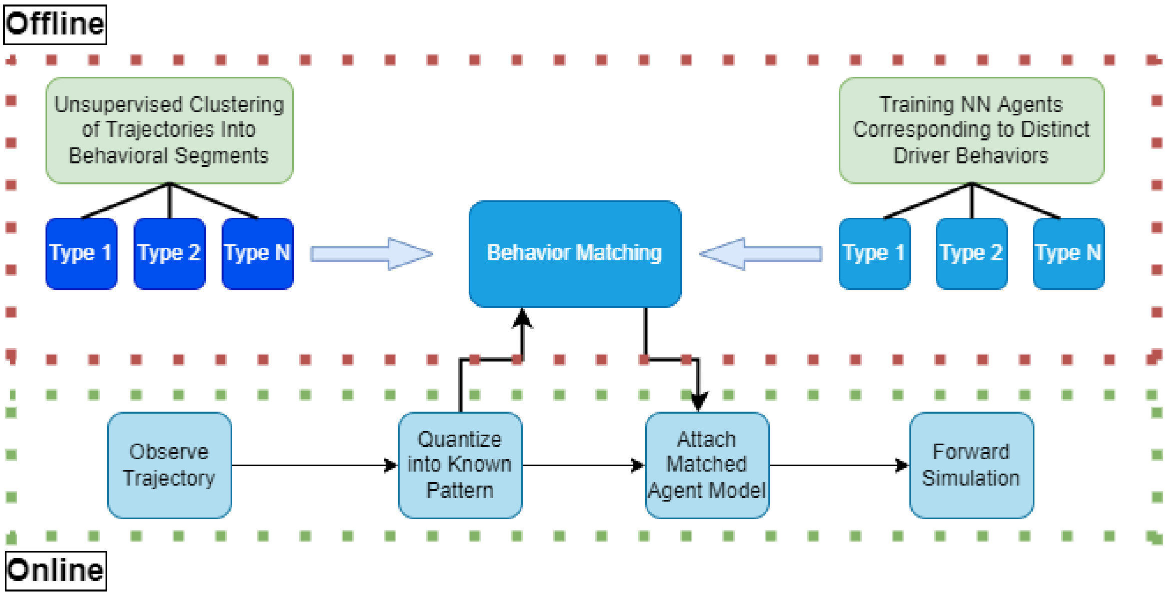

Figure 3.1. The architecture of the proposed system. An offline process of trajectory clustering to find different behavioral patterns is done using the dataset. On the other side, NN agents are trained corresponding to distinct driver behaviors and these are matched with the observed cluster center. Then the matched agent model is used in forward simulations. At inference time, an observed agent's trajectory is quantized into known patterns, behavioral matching module takes this input and outputs the corresponding NN Agent. The observed traffic scene is reconstructed in a simulation environment and now this scene can be used in AV testing.

3.1. Training of Different Behavior Agents

Agents are modeled by implementing the training strategy explained in [6]. In this work, there are two ways of creating different behavioral agents. The first variability is changing the level of reasoning a driver model has. A zero-level driver is only reactive and does not change lane. 1-level driver is a model that takes into account what the other driver will do, and 2-level driver has a two-level depth in action planning. The second variability of behavior comes from selecting different rewards for agents. Different reward parameters correspond to different behavioral agents.

The reward equation is,

$$R = w_1.c + w_2.s + w_3.d + w_4.e, \tag{3.1}$$

where $w_1, ..., w_4$ are driver-specific weights. The rest of the terms are:

- c: occurrence of collision. -1 or 0.
- s: difference between the speed of the driver. and the mean speed normalized by the maximum speed. [0,1].
- d: headway penalty. -1 if driver gets too close.
- e: penalty on effort consumption. Or comfort parameter. -1,-0.5,0.

By changing reward parameters, it is possible to train agents with different behavioral patterns. More discussion is provided in the experiments chapter.

3.2. Trajectory Encoding

Meaningful behavior extraction can be better accomplished if the trajectory data is represented in a way that selects relevant portions of the trajectory. In highway study, there is ample information about each vehicle; *position, velocity, acceleration, lane number, height, width, surrounding vehicles.* In [20], the trajectory is represented without interaction, using velocity and acceleration of each GPS trajectory. Other than a trajectory, dynamic traffic graph representation for behavior extraction has been studied in [33], which accounts for neighbouring vehicles. Combining the two approaches for highway trajectory case, two different trajectory encoding methods for highway data has been proposed.

The first formulation is a trajectory encoding that does not take other vehicles into account. Here the aim is to use this encoding to find 2D behavioral patterns without interaction, validate the trajectory representation. Then this validation will be the basis for an improved, interaction aware trajectory encoding.

Standard, non-interaction trajectory formulation is,

$$[V_x, V_y, A_x, A_y]. \tag{3.2}$$

The second formulation takes interaction into account. The distances of each vehicle, as defined by 3.4, are concatenated to the first trajectory representation, in 3.2. In Figure 3.2, the surrounding vehicles are represented. The formulation can be seen in equation 3.3. R, L, A, B, F are short for Right, Left, Along, Back and Front, respectively. Interaction aware fomulation is,

$$[V_x, V_y, A_x, A_y, d(F), d(B), d(RF), d(RB), d(RA), d(LF), d(LB), d(LA)], \tag{3.3}$$

where the distance, $d()$, is defined as,

$$d(OtherVehicle) = \begin{cases} e^{-d(y_{other}, y_{ego})}, & \text{if } \textit{exists} \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

3.3. Clustering into Behavioral Segments

After encoding trajectories, clustering them into different behaviors is the next challenge. The aim is to find behavioral segments that are distinct, semantically meaningful, and corresponding to a set of action in highway driving. The meaning of what a behavioral segment is based on the application. In trajectory prediction literature, this may mean an action of 3 to 5 seconds. Following the logic that a lane change takes about 5 seconds from the dataset [8], 5 seconds is chosen as the length of behavioral segments to be identified and modeled.

A set of challenges arise when highway driving is to be clustered into behavioral segments. Clustering method and distance metric play an important role in the results of behavioral segments. Here, a clustering method and distance that has been proved to be useful in vehicle trajectory clustering has been borrowed from literature. A formulation similar to [20] has been developed. Where a Self Organizing Map(SOM) [34] is trained to better define distances between trajectories. Then the output of SOM with each trajectory is used to perform clustering. Here, several different clustering methods have been tested, including kMeans Clustering and Agglomerative Clustering [35]. SOM + kMeans have shown to out perform the tested approaches in terms of clustering evaluation metrics.

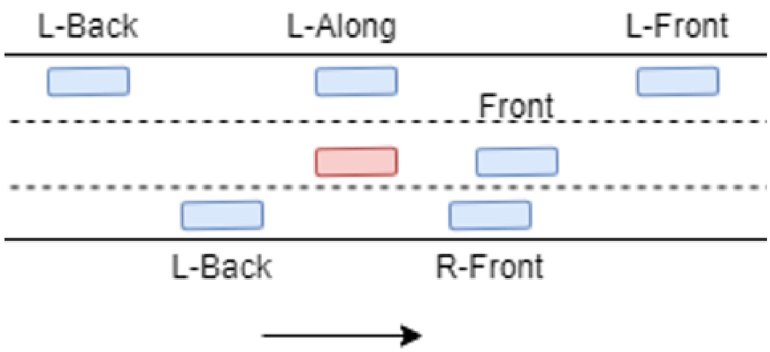

Figure 3.2. Vehicles surrounding an *Ego Vehicle* on highway. For each vehicle, all possible neighbors are defined. The feature from each neighbour is the distance from 3.4.

3.4. Behavior Matching

Clusters of trajectories corresponding to different behavioral segments are obtained, and NN driven simulated agents with different behavioral patterns have been trained. The next step in the formulation is to match these segments with NN agents. In order to match the observed trajectory with a simulated agent, the patterns of the simulated agent and the trajectory of the simulated agent need to be matched. In the literature, such matching problems have been addressed by [36] and [37]. In the cases where the model is controlled by a parameter, a parameter search method using particle filter can be used. When there is a discrete matching problem, then trajectory match and maximum entropy match method can be used. In this work, a trajectory match has been used to match the observed pattern to simulated agent pattern.

Require Trajectory (x_i, y_i). Models $(R_1, ..., R_N)$
Output Best matching model parameters
initialization;
for Every trajectory in cluster c_i **do**
    for Every model R_j **do**
        Calculate trajectory match score;
    end for
    j with maximum score;
    Cluster[i][j]+=1;
end for
return cluster model match list.

Figure 3.3. Behavior Matching Algorithm-1.

In the application domain, the observed trajectory length may differ from the identified behavioral segments. Therefore, two algorithms have been proposed to match the observed behavior with the trained agent models, Figure 3.4 and Figure 3.5.

3.5. Inference & Simulation

In inference time, the goal is to observe a vehicle for a short period of time *observation time*, then match a behavioral model to that vehicle and reconstruct a traffic scene with simulated agents. Once a traffic scene is reconstructed, it can be used for simulation, prediction and AV testing.

It's worth noting that at this point, all simulated agents are trained and a behavioral pattern of each agent is matched with the observed dataset trajectories. The remaining problem becomes assigning an observed trajectory into a known trajectory. After selecting the cluster centers and representative trajectories from the large dataset, the observed trajectory is assigned to one of the known trajectory patterns. Since the NN Agent corresponding to that cluster has been matched before, now it is possible to reconstruct traffic scene in simulation.

Different use cases and performance metrics are reported in the experiments section. Since this is a novel approach, its evaluation is also novel. Trajectory prediction results and lane change modeling results have been proposed to assess the quality of this step.

Require Short trajectory (x_i, y_i). Longer trajectory cluster centers $(C_1, ..., C_k)$
Output Best matching cluster center.
initialization;
for Every trajectory in cluster c_i **do**
  Use a moving window to calculate the match score;
  Cluster[i][j]+=1;
end for
return Cluster center with the highest match score.

Figure 3.4. Behavior Matching Algorithm-2.

4. EXPERIMENTS AND RESULTS

The study of driver behavior modeling does not have a standard evaluation in the literature, as the application domains are diverse. In this section, a formulation related to traffic scene reconstruction has been proposed. The logic behind this is its importance in AV testing. AVs are tested in the real world for a limited time and tested in simulations for the majority of the time. After recording real-world tests, the same scenes are reconstructed and tested with variations. An example of this approach is demonstrated by a study from Waymo, [38]. Traffic accidents have been recovered and Waymo AV has been tested against these scenarios.

The main proposal of reconstructing a traffic scene put forth by this thesis has several steps to it. First clustering of behavioral segments are created, then matched with simulated agents and used in the reconstructed simulations. Each step of the method requires well-designed experiments to evaluate how well that task is achieved. Therefore clustering, matching and inference performance should be analyzed separately.

4.1. Methodology

The main methodology for testing the performance of the proposed approach is demonstrated in the Figure 4.1. The scene is reconstructed and simulation is carried on. In the end, the resulting behavior is compared with the dataset behavior. Then this reconstruction error is analyzed in its ability to predict the vehicle trajectory and lane change.

Aside from the main evaluation, some intermediate evaluations are also needed to assess behavioral clustering quality. There are several approaches in the literature to assess separability and between likeliness of clusters. These metrics are accompanied with semantic evaluation for the sake of making more sense of the clusters and behavioral patterns.

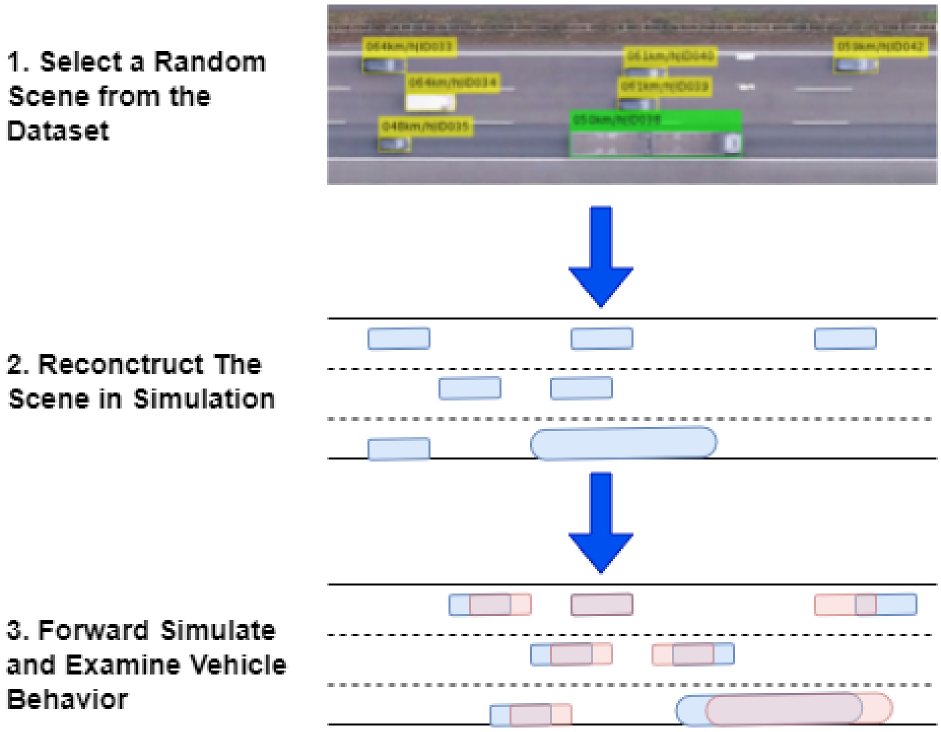

Figure 4.1. Main experimentation method. The top picture belongs to the traffic scene from the dataset. The blue boxes represent constructed vehicles in simulation. The red boxes represent the actual positions of the vehicles from the dataset, after some observation time.

4.2. Dataset

The dataset chosen for experiments is HighD [8], which is published under the title: "A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems". It's a collection of trajectories of vehicles, captured at 25 Hz. The attributes related to each vehicle are:

- X, Y positions.
- X, Y velocities.
- X, Y accelerations.
- Lane id.
- Neighbour vehicle ids.
- Headway, the distance to the preceding vehicle.

Total recordings of 16.4 hours are divided by 100 recordings with different day times and 6 different locations. The distribution of trucks and cars is mentioned in Table 4.1. In the publication, it is mentioned that there has been care to put interesting traffic scenarios in the recordings. This means that there is some interaction in almost all of the scenes in the dataset, instead of empty road recordings.

Table 4.1. Detailed information about the dataset.

HighD Dataset Specifications	
Duration of Recordings [hours]	16.4
Lanes per Direction	2-3
Recorded Distance [m]	400-420
Number of Vehicles	110 000
Number of Cars	90 000
Number of Trucks	20 000
Distinct Locations	6
Distinct Recordings	100

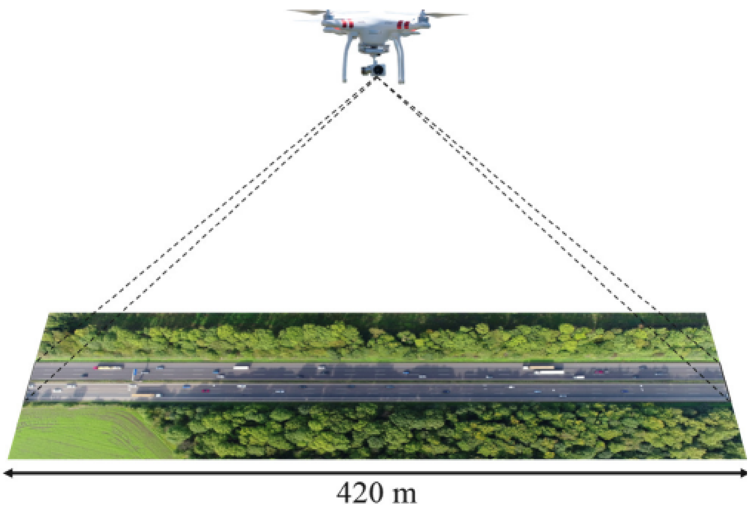

Figure 4.2. HighD dataset capturing method.

4.3. Performance Metrics

4.3.1. Traffic Reconstruction

There is no agreed method of measuring performance in driver behavior modeling, as it is with the main proposed method. In the scope of this work, motion prediction after scene reconstruction has been used as a method to measure performance. Given an observation of length p frames, the motion prediction capability of the system is measured on n frames. There are a variety of experiments ranging from 1-8 seconds of observation and prediction lengths. Success in traffic reconstruction can be measured by Average Displacement Error(ADE) and Final Displacement Error(FDE) metrics.
Average Displacement Error: The root mean square error (RMSE) over prediction next n frames,

$$\frac{1}{n}\sum_{j=t+1}^{t+n}(\hat{x}^j - x^j)^2. \tag{4.1}$$

Final Displacement Error: The absolute difference between final prediction position and predicted position,

$$|\hat{x}^{t+n} - x^{t+n}|. \tag{4.2}$$

Another component of traffic reconstruction success metric is lane change prediction success. It is of critical importance in highway setting that simulated models does perform lane change at certain frequency, in line with the data. To address the success of models in prediction lane change, specific scenarios have been selected, in which certain vehicles do perform lane change in the observation time. Then the ratio of successfully modeling the lane change for selected vehicles is reported.

4.3.2. Trajectory Clustering

Clustering performance metrics are various in the literature. The reasoning behind which to use should be about the kind of data that is being used in clustering.

Following the work of [20], two clustering success metrics have been used to analytically determine success of the proposed method against others in the literature.

Calinski-Harabatz [39] is a very popular clustering performance metric. It gives higher scores for well separated clusters. One drawback of this method is that it does not work well with density based clustering methods. But for the sake of trajectory clustering, it is a good indicator of between cluster separability.

Davies-Boulding [40] is another widely used clustering performance metric. This metric is especially useful for within cluster similarity measurement. It is also very simple to calculate. The drawback of this metric is that it only uses point wise distances which is susceptible to distance measurement method of the dataset.

Lastly, clustering performance can be measured by semantically labeling the clusters. This measurement is used as a guide, rather than an analytic success metric. Each cluster's average acceleration and velocity values have been taken into account as well as their trajectory in a trajectory plot. Such a semantic meaning is inspired from [20].

4.4. Experiments

4.4.1. Traffic Reconstruction: Trajectory Prediction

An example reconstructed scene is depicted in Figure 4.3. The traffic is reconstructed and the final displacement error is shown in the figure after 5 seconds of prediction. In the process, vehicles are matched with the corresponding pre-trained NN agents after 1 second observation, and the resulting simulation is shown after 5 seconds of prediction. The initial locations and speeds are taken from a random scene from the dataset.

Here, the task is to predict the locations of the vehicles. Vehicles are observed for n seconds and their locations after p seconds are predicted. Commonly n is between 1-3 seconds and p is between 3-8 seconds. It is desired to predict longest time horizon with the shortest possible observation time. The solution presented in this work is compared against SOTA trajectory prediction methods. In cases where observation is shorter, the proposed method outperforms compared implementations.

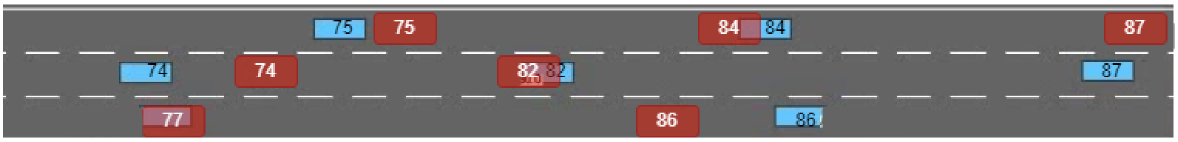

Figure 4.3. Example result of reconstruction and prediction. The blue boxes represent simulated vehicles. The red boxes represent where the final place would be in real traffic data. It can be observed that some vehicles are predicted well, like #77, but some are not, like #87. There is a lane change prediction for #87 that does not hold.

Table 4.2. Final prediction results on HighD dataset reconstruction task.

Method	**3-3 ADE**	**3-6 ADE**	**1-8 ADE**
Trajectory Prediction (GRIP)	**1.56**	4.19	8.98
Traffic Simulation with RL + Random Assign	4.15	5.85	9.13
Traffic Simulation + MatchALG1	2.84	**4.17**	**7.29**
Traffic Simulation + MatchALG2	2.64	4.69	7.30
Constant Velocity	1.81	4.99	11.98

4.4.2. Comparison Method

A trajectory prediction algorithm has been implemented with the same dataset to compare the results of proposed method. GRIP [32], Graph Convolutional Trajectory Prediction, uses GCNs to predict the traffic. It is the SOTA in highway trajectory prediction at time of writing. The model is designed for 3 seconds observation and 3 seconds prediction. It fails to capture any relations longer than that time. It also fails to model any outside-lane modeling as shown in 4.4.

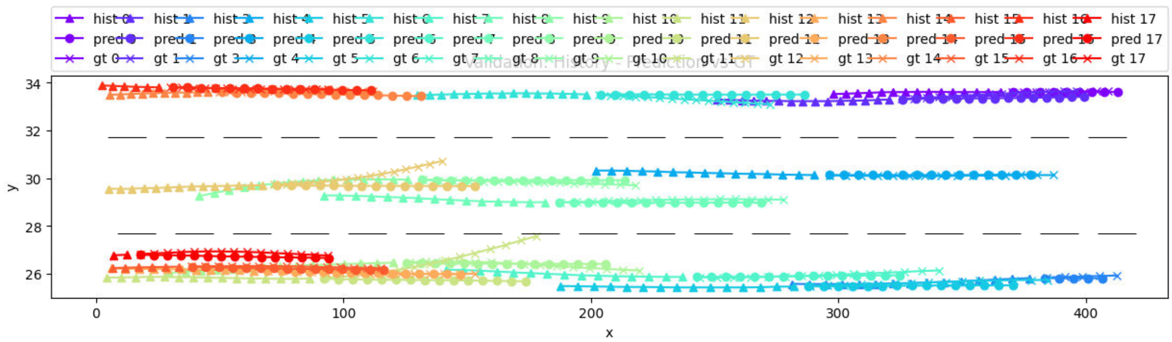

Figure 4.4. Comparison method GRIP [32] in reconstructing a scene. The triangles represent past, the dots represent predictions and x symbols represent future locations. It can be seen that the model is incapable of modeling uninitiated lane change maneuvers, and fails at predicting any outside lane movement. For example, yellow vehicle at the middle left of the scene is changing lane in the future, but the prediction continues smoothly. This model does not have the capability of adding rich maneuvers.

The performance of proposed approach has been tested against comparison method with different 3 different setups:

- *3-3*: 3 seconds observe and 3 seconds predict.
- *2-6*: 2 seconds observe and 6 seconds predict.
- *1-8*: 1 second observe and 8 seconds predict.

In the end result, the proposed method models the mid-term better than trajectory prediction (TP) method but fails to match correct behavior in the short observation of 1 seconds. There has to be improvements in agent training and matching of observed trajectory to agents in short observations. It is also worth noting that comparison method is implemented manually, as the code for the work is not published.

4.4.3. Traffic Reconstruction: Lane Change Modeling

One important aspect of successful traffic reconstruction is the ability of the reconstructed scene to perform lane change actions. In the comparison method, GRIP, the trajectories are predicted quite successfully however, there is no logic of *lane-change* or behavioral highway actions. In fact, the approach cannot recover scenarios where vehicles are in the middle of the lane change, as can be seen in Figure 4.5.

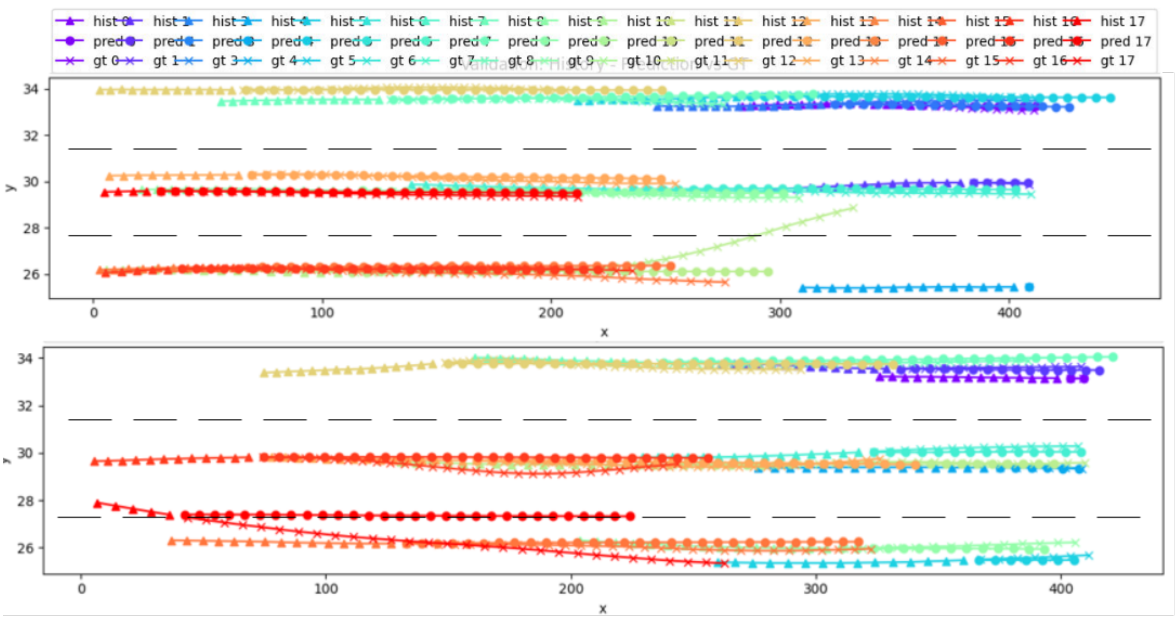

Figure 4.5. Comparison method lane change modeling. The figure is from two selected scenes from the dataset, top and down. X is lateral change and Y is longitudinal distances in meters. Each color represents a different vehicle. Triangles are past, crosses are real and circles are predicted trajectory. In the top plot, the model fails to predict a lane change that occur at the end of prediction window. The bottom plot is an example scene where lane change happens in the middle of observation window. In both cases comparison method fails to incorporate the logic of lane change.

The success criteria is the ability of each model to correctly predict the next lane of the selected vehicles. Out of the 30 scenarios, the proposed approach can model 14 of the lane changes, whereas the comparison method can only capture 5 of the lane changes. Table 4.3 shows the results for each of the cases. The lane change scenarios were randomly picked, as they were not annotated in the dataset. This approach of selecting certain scenes manually can be seen in other works in the literature [41].

4.4.4. Clustering With No-Interaction

As mentioned previously, no clustering method that accounts for interaction have been studied in the literature. Therefore, an intermediate step of using suggested clustering method without interaction have been developed and analyzed. Since it is an intermediary analysis, only semantic analysis has been provided. The visualization can be seen in Figure 4.6, and the corresponding semantic explanations are reported in Table 4.4. The number of clusters without interaction has been chosen as 6, following the logic that left, right lane change, along with weaving and lane follow behavior make the number reasonable.

Table 4.3. Lane change modeling results on the selected scenarios from HighD dataset. In this experiment, certain vehicles start the scene at some lane, and finished the scene in another lane. 3-6 and 1-8 refer to observation time and prediction time, respectively.

Method	**3-6 ADE**	**1-8 ADE**
Trajectory Prediction (GRIP)	5/30	1/30
Traffic Simulation with RL + Random Assign	6/30	6/30
Traffic Simulation + MatchALG1	12/30	11/30
Traffic Simulation + MatchALG2	**14/30**	**11/30**
Constant Velocity	5/30	1/30

The dataset used in the literature of vehicle trajectory clustering is some linear vehicle data that does not account for the infrastructure or the interaction between vehicles. Still, there is value in performing such clustering methods with the available data. The goal of this intermediary step is to make sure that the clustering methods in the literature give meaningful results in highway data.

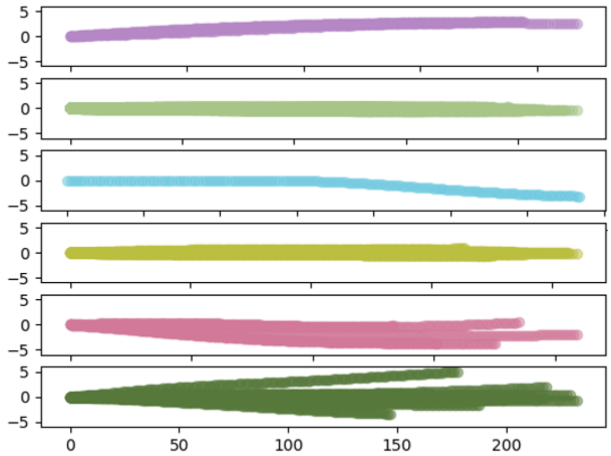

Figure 4.6. Cluster visualizations without interaction encoding. Clusters from 1-5 have very distinct behavioral features, when cluster 6 has no clear semantic explanation.

Semantic meanings from a GPS trajectory of city driving does not always find correspondence in highway driving. This is why in this work, several of the behaviors are manually selected. Such as right lane change, left lane change, high acceleration behavior. Another selected behavior is weaving. Weaving means the vehicle recklessly moving between lanes. This movement usually correspond to aggressive driving. There are more semantic labels in the reference paper, but they are not included as they are not applicable to highway driving.

4.4.5. Clustering with Interaction

Adding interaction behavior has been discussed in Chapter 3. In its essence, it requires additions of nearby vehicles' distances to the trajectory representation used in non-interaction representation. The selection of K, i.e. number of clusters, becomes much more complex in the case of interactive behaviors. In principle, interaction adds one dimension to all the non-interactive behaviors; *cooperativeness*. When interaction is added, a *left lane change* behavior can be either of *cooperative* or *aggressive*. The same goes for all the behavioral segments previously defined. Therefore, the number of clusters becomes twice the non-interactive behaviors, 12.

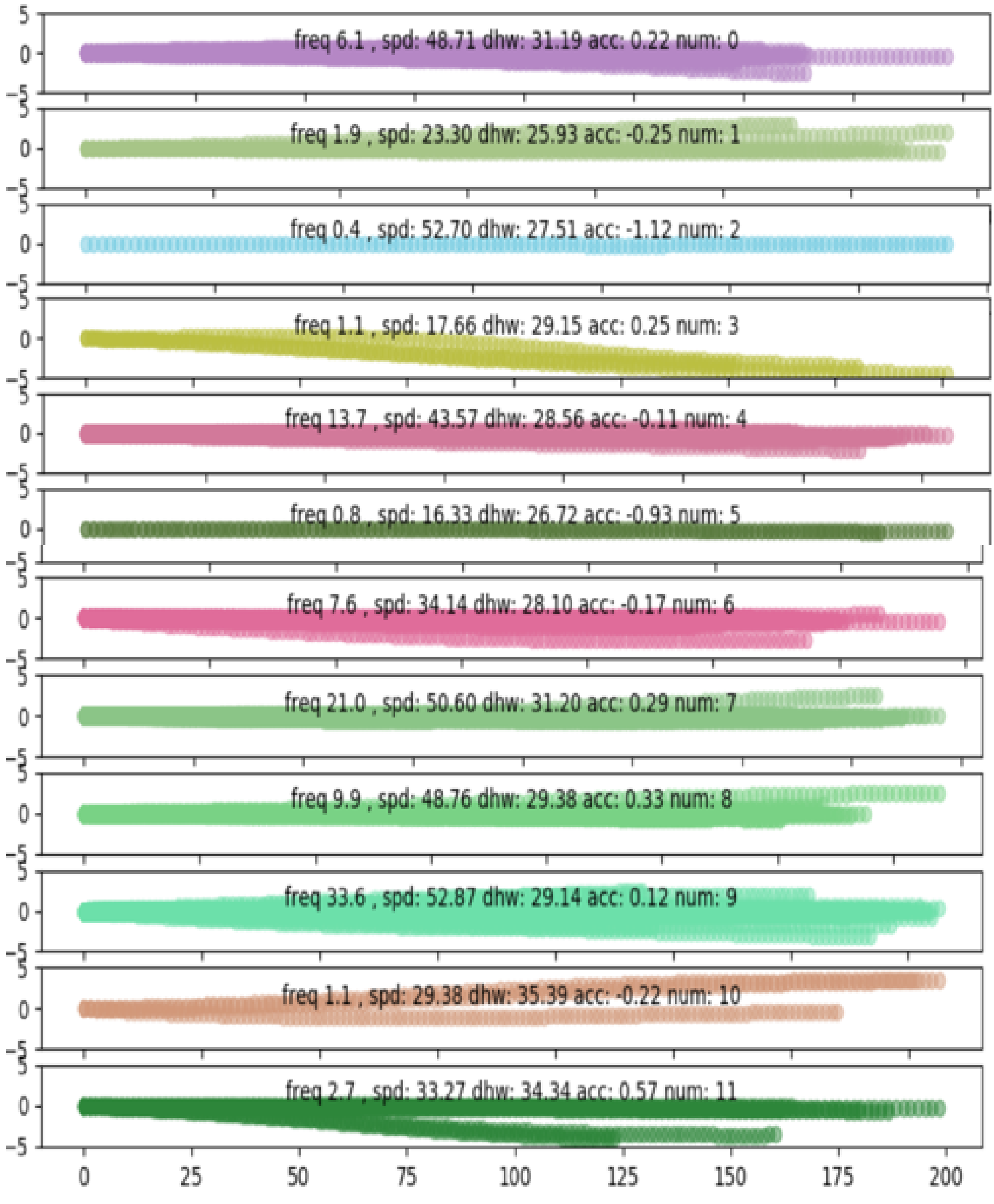

Figure 4.7. Cluster visualizations with interaction encoding. Frequency, average speed, average distance headway(dhw), average acceleration(acc) and cluster number(num) have been denoted on each row of the clusters.

Creating semantic meanings from interactive vehicle trajectory data is never done in the literature. In this work, such an attempt has been done. Even in the non-interactive vehicle trajectory clustering literature, most of the times semantic results of the clusters are not provided. This may be due to semantic labels not being always consistent, even clusters not always being separable from one another. The motivation in this work is to provide the results, even though several of the clusters cannot be semantically separated. This seems to be a limitation of the current literature in this domain.

In Figure 4.7 clusters can be observed. For some clusters, the main factor of differentiation is speed, such as clusters 4, and 5. In some clusters the differentiation comes from lateral movement, such as cluster 3 and 10. As with the non-interaction version, it can be shown that there is at least one cluster that is occurring very frequently, cluster number 9. Its occurrence frequency is %33.6, which is very high for a clustering of 12 categories. It can bee seen that these trajectories, in cluster number 9, all have very high speed averages. The same phenomenon has been observed in non-interaction behaviors. So it can be concluded that the used clustering technique has some weak points.

Table 4.4. Semantic meanings of no-interaction clusters. The labels are taken from examples in the literature, [20].

Explanation of Cluster	**Cluster ID**
Left lane change	1
Driving at normal speed	2
Right lane change	3
High acceleration behavior	4
Weaving at high speed	5
Weaving at medium speed	6

Table 4.5. Semantic meanings of interaction clusters.

Frequency	**AvgSpeed**	**AvgHeadway**	**AvgAcc.**	**Explanation**	**ID**
6.1	48.71	31.19	0.22	High Speed. Constant Velocity	0
1.9	23.3	25.93	-0.25	Slow Side Slipping.	1
0.4	52.7	27.51	-1.12	High Acc. Lane Keep	2
1.1	17.66	29.15	0.25	Aggressive. Right Lane Change	3
13.7	45.57	28.56	-0.11	Medium Speed. Constant Velocity	4
0.8	16.33	26.72	-0.93	High Acceleration Behavior.	5
7.6	34.14	28.1	-0.17	Medium Speed. Swearving.	6
21.0	50.6	31.2	0.29	Side Slipping	7
9.9	48.76	29.38	0.33	Side Slipping	8
33.6	52.87	29.14	0.12	High Speed.	9
1.1	29.38	35.39	-0.22	Cooperative Left Lane Change	10
2.7	33.27	34.34	0.57	Cooperative Right Lane Change	11

4.4.6. Clustering Success Metrics

Clustering success metrics are good indicators of consistency, but they are far from giving exact semantic success results. Following the literature on vehicle trajectory clustering, Calinski-Harabatzs [39] and Davies-Bouldin [40] scores have been reported as part of clustering analytical success metrics.

The proposed method of using SOM+kMeans have been compared against Hierarchical Cluster Analysis, (HCA) [35]. Following the work of Camarer et.al. [42], humans are hierarchical thinkers and therefore using HCA in human behavior analysis is meaningful.

In Table 4.6, interaction encoding and non-interactive encoding have been compared with different number of cluster numbers, k. Calinski score is better when it is higher, Davies score is better when it is smaller. It can be shown analytically that SOM + kMeans have been the better clustering method.

The results of analytical evaluation, together with semantic evaluation show that there is a value in clustering interactive vehicle trajectories, however current approaches are not perfect. They have some shortcoming that need to be addressed. Yet, using these clusters in behavioral identification increases performance in data validated traffic simulation. It can be concluded that vehicle trajectory clustering is useful as it is but needs further work to clearly identify driving patterns.

Table 4.6. Clustering performance metrics.

Method	**k**	**Interaction**	**Davies**	**Calinski**
HCA	5	No	2.631	44.536
SOM + kMeans	5	No	1.864	53.262
HCA	7	No	2.363	40.023
SOM + kMeans	7	No	1.385	55.822
HCA	12	Yes	1.756	113.547
SOM + kMeans	12	Yes	**0.889**	**153.285**

5. CONCLUSION

There is great value in modeling human driving in simulations. This work has provided a method of using different behavioral patterns in an AV simulation environment for testing and verification purposes. It has a long way of modeling an in-depth driver, but it is a step forward. This work makes an important contribution in providing a set of tools to reconstruct a driving dataset scene in simulation. These simulations can be further used by AV researchers.

Currently, the behavior matching method is not superior in its motion prediction feature to trajectory prediction methods. But it creates an environment suitable for simulations, based on strict vehicle models and drivers. There have to be improvements in observation to behavior matching features. As well as NN agent training.

Long-term driving behavior (longer than 30 seconds) has not been discussed in this work because of a lack of data. There is a need for observation of a single agent for a longer time and work on improving the current understanding of drivers in different situations.

The tools available to the online community are very diverse but not connected. There is no tool to reconstruct a dataset in a simulation. This work makes a contribution on that front. However, the current approach is very specific to a single simulation environment and a single dataset. There need to be inter-simulation communication protocols so that all different simulation platforms can use certain scenes described by one of them.

Another future work can be done by addressing aggressiveness and driving patterns. This would require a collection of a new dataset. This kind of work would have a physiological study of people and their driving recording. The data could be collected using a simulator or real-world driving with sensory recordings. There are some studies on this front however none of them have shared their data publicly. Any new published dataset, such as this one, should be compatible with the most popular online tools to analyze this data.

REFERENCES

1. Hendricks, D. L., J. C. Fell and M. Freedman, "The Relative Frequency of Unsafe Driving Acts in Serious Traffic Crashes", *NHSA Technical Report 1700.7*, Vol. 1, p. 7, 2001.

2. Administration, N. H. T. S., "Traffic Safety Facts", *Traffic Safety Facts 2015*, p. 101.

3. Schneider, J., *Why People Keep Rear-Ending Self-Driving Cars*, 2018, `https://www.wired.com/story/self-driving-car-crashes-rear-endings-why-charts-statistics/`, accessed in Jan 2022.

4. Brown, K., K. Driggs-Campbell and M. J. Kochenderfer, "A Taxonomy and Review of Algorithms for Modeling and Predicting Human Driver Behavior", *arXiv Preprint arXiv:2006.08832*, 2020.

5. Kuhn, H. W., "11. Extensive Games and the Problem of Information", *A Value for N-Person Games*, 1953.

6. Albaba, M. and Y. Yildiz, "Driver Modeling through Deep Reinforcement Learning and Behavioral Game Theory", *arXiv Preprint arXiv:2003.11071*, 2020.

7. Hu, Y., A. Nakhaei, M. Tomizuka and K. Fujimura, "Interaction-aware Decision Making with Adaptive Strategies under Merging Scenarios", *arXiv Preprint arXiv:1904.06025*, 2020.

8. Krajewski, R., J. Bock, L. Kloeker and L. Eckstein, "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems", *arXiv Preprint arXiv:1810.05642*, 2018.

9. Michon, J. A., "A Critical View of Driver Behavior Models: What Do We Know,

What Should We Do", L. Evans and R. C. Schwing (Editors), *Human Behavior and Traffic Safety*, pp. 485–524, Springer US, Boston, MA, 1985.

10. Lefèvre, S., D. Vasquez and C. Laugier, "A Survey on Motion Prediction and Risk Assessment for Intelligent Vehicles", *ROBOMECH Journal*, Vol. 1, No. 1, p. 1, 2014.

11. Lopez, P. A., M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner and E. Wiessner, "Microscopic Traffic Simulation using SUMO", *21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, 2018.

12. Derbel, O., T. Peter, H. Zebiri, B. Mourllion and M. Basset, "Modified Intelligent Driver Model for Driver Safety and Traffic Stability Improvement", *IFAC Proceedings Volumes*, Vol. 46, No. 21, pp. 744–749, 2013.

13. Treiber, M., A. Hennecke and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations", *Physical Review E*, Vol. 62, No. 2, pp. 1805–1824, 2000.

14. Shen, J. and X. Jin, "Detailed Traffic Animation for Urban Road Networks", *Graphical Models*, Vol. 74, pp. 265–282, 2012.

15. Alizadeh, A., M. Moghadam, Y. Bicer, N. K. Ure, U. Yavas and C. Kurtulus, "Automated Lane Change Decision Making using Deep Reinforcement Learning in Dynamic and Uncertain Highway Environment", *arXiv Preprint arXiv:1909.11538*, 2019.

16. Oyler, D. W., Y. Yildiz, A. R. Girard, N. I. Li and I. V. Kolmanovsky, "A game theoretical Model of Traffic with Multiple Interacting Drivers for use in Autonomous Vehicle Development", *American Control Conference (ACC)*, pp. 1705–1710, 2016.

17. Yoo, J. and R. Langari, "A Stackelberg Game Theoretic Model of Lane-Merging",

arXiv Preprint arXiv:2003.09786, 2020.

18. Sutton, R. S. and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 2018.

19. Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, “Playing Atari with Deep Reinforcement Learning”, *arXiv Preprint arXiv:1312.5602*, 2013.

20. Siami, M., M. Naderpour and J. Lu, “A Mobile Telematics Pattern Recognition Framework for Driving Behavior Extraction”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 3, pp. 1459–1472, 2021.

21. Wang, Z., X. Liao, C. Wang, D. Oswald, G. Wu, K. Boriboonsomsin, M. Barth, K. Han, B. Kim and P. Tiwari, “Driver Behavior Modeling Using Game Engine and Real Vehicle: A Learning-Based Approach”, *IEEE Transactions on Intelligent Vehicles*, Vol. 5, pp. 738–749, 2020.

22. Besse, P., B. Guillouet, Brendan Guillouet, J.-M. Loubes and F. Royer, “Review and Perspective for Distance-Based Clustering of Vehicle Trajectories”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 11, pp. 3306–3317, 2016.

23. Harmening, N., M. Biloš and S. Günnemann, “Deep Representation Learning and Clustering of Traffic Scenarios”, *arXiv Preprint arXiv:2007.07740*, 2020.

24. Atev, S., G. Miller and N. P. Papanikolopoulos, “Clustering of Vehicle Trajectories”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 3, pp. 647–657, 2010.

25. Wang, W., A. Ramesh and D. Zhao, “Clustering of Driving Encounter Scenarios Using Connected Vehicle Trajectories”, *IEEE Transactions on Intelligent Vehicles*, Vol. 5, No. 3, pp. 485–496, 2020.

26. Liu, H., T. Taniguchi, Y. Tanaka, K. Takenaka and T. Bando, "Visualization of Driving Behavior Based on Hidden Feature Extraction by Using Deep Learning", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 9, pp. 2477–2489, 2017.

27. Administration, N. H. T. S., *NGSIM Next Generation Simulation US Highway 101 Dataset*, 2007, `https://www.fhwa.dot.gov/publications/research/operations/07030/`, accessed in Sep 2021.

28. Zhan, W., L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps", *arXiv Preprint arXiv:1910.03088*, 2019.

29. Chang, M.-F., J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan and J. Hays, "Argoverse: 3D Tracking and Forecasting with Rich Maps", *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

30. Houston, J., G. Zuidhof, L. Bergamini, Y. Ye, A. Jain, S. Omari, V. Iglovikov and P. Ondruska, *One Thousand and One Hours: Self-Driving Motion Prediction Dataset*, 2020, `https://level-5.global/level5/data`, accessed in Sept 2021.

31. Sun, P., H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in Perception for Autonomous Driving: Waymo Open Dataset", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020.

32. Li, X., X. Ying and M. C. Chuah, "GRIP++: Enhanced Graph-Based Interaction-Aware Trajectory Prediction for Autonomous Driving", *arXiv Preprint arXiv:1907.07792*, 2020.

33. Chandra, R., T. Guan, S. Panuganti, T. Mittal, U. Bhattacharya, A. Bera and D. Manocha, "Forecasting Trajectory and Behavior of Road-Agents Using Spectral Clustering in Graph-LSTMs", *arXiv Preprint arXiv:1912.01118*, 2020.

34. Saraee, M., S. Moosavi and S. Rezapour, "Application of Self Organizing Map (SOM) to Model a Machining Process", *Journal of Manufacturing Technology Management*, Vol. 22, No. 6, pp. 818–830, 2011.

35. Szekely, G. J. and M. L. Rizzo, "Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method", *Journal of Classification*, Vol. 22, No. 2, 2005.

36. Schwarting, W., A. Pierson, J. Alonso-Mora, S. Karaman and D. Rus, "Social Behavior for Autonomous Vehicles", *Proceedings of the National Academy of Sciences*, Vol. 116, No. 50, pp. 24972–24978, 2019.

37. Bhattacharyya, R., R. Senanayake, K. Brown and M. Kochenderfer, "Online Parameter Estimation for Human Driver Behavior Prediction", *arXiv:2005.02597*, 2020.

38. Scanlon, J. M., K. D. Kusano, T. Daniel, C. Alderson, A. Ogle and T. Victor, "Waymo Simulated Driving Behavior in Reconstructed Fatal Crashes within an Autonomous Vehicle Operating Domain", *Accident Analysis & Prevention*, Vol. 163, p. 106454, 2021.

39. Caliński, T. and H. JA, "A Dendrite Method for Cluster Analysis", *Communications in Statistics - Theory and Methods*, Vol. 3, pp. 1–27, 1974.

40. Davies, D. L. and D. W. Bouldin, "A Cluster Separation Measure", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-1, No. 2, pp. 224–227, 1979.

41. Huang, Z., C. Lv and J. Wu, "Modeling Human Driving Behavior in Highway

Scenario using Inverse Reinforcement Learning", *arXiv Preprint arXiv:2010.03118*, 2020.

42. Camerer, C. F., T.-H. Ho and J.-K. Chong, "A Cognitive Hierarchy Model of Games", *The Quarterly Journal of Economics*, Vol. 119, No. 3, pp. 861–898, 2004.