©Copyright 2022
Jundi Liu

Toward Trust-calibrated Customized Vehicle Automation

Jundi Liu

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:
Ashis G. Banerjee, Chair
Linda Ng Boyle, Chair
Archis Ghate
Maya Cakmak

Program Authorized to Offer Degree:
Industrial & Systems Engineering

University of Washington

Abstract

Toward Trust-calibrated Customized Vehicle Automation

Jundi Liu

Co-Chairs of the Supervisory Committee:
Ashis G. Banerjee
Industrial & Systems Engineering and Mechanical Engineering

Linda Ng Boyle
Industrial & Systems Engineering and Civil & Environmental Engineering

Vehicle automation has long been an essential component of modern vehicular technologies. From the Advanced Driver Assistance System (ADAS) to fully autonomous vehicles, these technologies are trying to enhance road safety and improve the driving experience. However, their acceptance does not seem to keep up with the fast-growing market penetration. There have been many attempts to improve the acceptance of vehicle automation. The most widely adopted methods are customization and trust calibration. However, to the best of our knowledge, none of the current studies provides a holistic framework to design a customized vehicle automation system that involves humans in the loop. This dissertation aims to propose a systematic paradigm to design a human-aware and trust-calibrated vehicle automation system. We start from the system side to build customized vehicle automation using driving demonstrations from naturalistic driving data. The procedure consists of identifying driving style using Multivariate Functional Principal Component Analysis (MFPCA) and clustering analysis. Then, using the clustering result, we apply the Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) algorithm to train the personalized vehicle automation system from demonstrations. The result shows the effectiveness of our proposed method. Next, from the human side, we develop a customized real-time trust prediction model based

on trust dynamics. “Confident” and “Skeptical” trust dynamics are identified from the trust levels during interaction with the vehicle automation. Furthermore, the State Space models for real-time trust prediction improve trust prediction performance for further trust calibration considerations. Finally, we integrate customized vehicle automation with the trust models using the human-in-the-loop method to achieve trust-calibrated customized vehicle automation. The proposed algorithm takes a step toward our goal of building human-aware customized automated systems. The findings have implications for future human-vehicle interaction design and trust calibration in vehicle automation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AI: Artificial Intelligence

NHTSA: National Highway Traffic Safety Administration

SAE: Society of Automotive Engineers

NDD: Naturalistic Driving Data

UMTRI: University of Michigan Transportation Research Institute

IRL: Inverse Reinforcement Learning

FDA: Functional Data Analysis

GMM: Gaussian Mixture Model

MFPCA: Multivariate Functional Principal Component Analysis

FPCA: Functional Principal Component Analysis

PCA: Principal Component Analysis

PC: Principal Component

RL: Reinforcement Learning

MDP: Markov Decision Process

MAXENT IRL: Maximum Entropy IRL

DEEPIRL: Maximum Entropy Deep Inverse Reinforcement Learning

SPMD: Safety Pilot Model Deployment

IRB: Institutional Review Boards

FPC: Functional Principal Component

FNN: Feed-forward Neural Network

GAIL: Generative Adversarial Imitation Learning

POMDP: Partially Observable Markov Decision Process

AR: Augmented Reality

LR: Linear Regression

SS: State Space

EKF: Extended Kalman Filter

MSE: Mean Squared Errors

HCI: Human-computer Interaction

HRI: Human-robot Interaction

CI: Confidence Interval

ANOVA: Analysis of Variance

KDE: Kernel Density Estimation

ACKNOWLEDGMENTS

Words cannot express my sincere gratitude to my two advisors, Dr. Linda Boyle and Dr. Ashis Banerjee, who have provided invaluable patience and unconditional support throughout this journey. I could not imagine a better way of taking on this chapter of my life if not under your supervision. To my advisor Dr. Linda Boyle, you are like a lighthouse that guided me in the right direction, especially in the darkest time. You are a role model I will always look up to in academia who showed me what is possible to pursue and achieve in my future care. To my advisor Dr. Ashis Banerjee, you are like the most reliable safety net behind my back, encouraging me to keep moving forward without fearing to fall. Thank you for being my compass when I was confused and lost my direction.

I could not have undertaken this journey without all my committee members, Dr. Franziska Roesner, Dr. Maya Cakmak, and Dr. Archis Ghate, who generously provided knowledge and expertise. I sincerely appreciate all the feedback and for taking the time to be on my committee.

Additionally, this endeavor would not have been possible without the inspiration, accompany, and friendship of all my fellow peers in the Human Factors and Statistical Modeling Lab and Scale-independent Multimodal Automated Real Time Systems (SMARTS) Lab. Thank you, Fiete Krutein, Tianshu Feng, Steven Hwang, Huizhong Guo, Haena Kim, Ning Li, Erika Miller, Xingwei Wu, Wei Guo, Xingjian Yang, and Ekta Samani.

Lastly, I would be remiss in not mentioning my loved ones - my parents and Yuqi Shi. Their belief in me has kept my spirits and motivation high during this process. I would also like to thank my cats, Celine and July, for all the entertainment and emotional support.

Chapter 1
INTRODUCTION

Over the past decade, Artificial Intelligence (AI) has achieved tremendous success in assisting all aspects of our daily lives. From small devices like smartphones to large devices like autonomous cars, they all leverage AI techniques in complex decision-making tasks to change our lives. Moreover, due to the agile advancement in AI, prospective AI agents are expected to play the role of teammates rather than replace humans.

Among all noteworthy assistive devices, autonomous cars are one of the most significant applications due to the safe-critical and essential usage scenario in our daily commutes. The field of vehicle automation is evolving rapidly. Although it will likely be many years until fully autonomous vehicles are publicly available in a wide range of conditions, different levels of vehicle automation technologies expedite the pace to our final destination. As defined by the Society of Automotive Engineers (SAE) [97], the following vehicle automation level system is the most commonly used standard while categorizing vehicle automation technologies.

- Level 0: No automation.

- Level 1: Driver assistance - The vehicle can control either steering or speed autonomously in specific circumstances to assist the driver.

- Level 2: Partial automation - The vehicle can control both steering and speed autonomously in specific circumstances to assist the driver.

- Level 3: Conditional automation - The vehicle can control both steering and speed autonomously under normal environmental conditions, but requires driver oversight.

- Level 4: High automation - The vehicle can complete a travel autonomously under normal environmental conditions, not requiring driver oversight.

- Level 5: Full autonomy - The vehicle can complete a travel autonomously in any environmental conditions.

Most of the current commercial vehicles have equipped with Level 1 automation, and some of the newer models have Level 2 automation. However, only a few cars can achieve level 3 conditional automation. Nevertheless, there is a lot of good news in the autonomous car industry, such as leading companies that are gradually authorized to conduct road tests on urban roads in major cities of the United States. Moreover, a recent report released by National Highway Traffic Safety Administration (NHTSA) stated that there were 38,824 people killed in motor vehicle crashes in 2020. Luckily, studies have shown that vehicle automation can improve road safety by reducing road injuries and fatalities caused by human errors [2]. However, there are also rising concerns in society. For instance, a fatal crash may still happen when autonomous agents control the cars. And most importantly, people still don't trust the system to hand over the complete controls to a black-box AI agent.

Despite the considerable potential and positive influence on how we travel and commute, researchers are still trying to fix the current issues that impede the adoption and wide usage of vehicle automation technologies. First, autonomous cars are vulnerable and have limited capabilities to handle unexpected events. Nassi *et al.* show that the Tesla Autopilot function is still vulnerable to phantom attacks. Second, a widely adopted consensus is that vehicle automation with an average driving style will not be satisfactory to all individual drivers. For user acceptance and trust, they should not only be safe and reliable but also provide a comfortable user experience [58]. However, comfortableness was perceived differently by different drivers. Whereas some drivers may prefer an aggressive driving style, others might prefer a more conservative driving style. Therefore, it's crucial to customize the vehicle automation functions based on the drivers' demonstrated driving styles. Third, the acceptance of vehicle automation doesn't seem to catch up with the growing market penetration, partially due

to the trust issues that affect the reliance on the technologies. Instead of blindly behaving as it was programmed to, it should have the ability to respond and change to human trust dynamics during the interaction. Therefore, studies should explore the effective integration method to achieve trust calibration in vehicle automation. However, before we can integrate the trust dynamics into the customized systems, one of the crucial steps is to accurately model trust during interaction without interrupting or intruding since trust is a hidden and implicit cognitive state..

1.1 Research Objectives and Goal

The overall goal of this dissertation is to take a step toward effective human-system integration in vehicle automation to design a human-aware and trust-calibrated customized vehicle automation that teammates with human drivers in complex driving tasks. To achieve our goal, we leverage the Human-in-the-loop method to address the following research objectives:

- *Research Objective 1: Customized vehicle automation to drivers' preferences and differences*

  This research objective aims to extract the preferences and differences underlying driving behavior and styles using Naturalistic Driving Data (NDD) and customize the vehicle automation to capture the identified driving styles. Hence, we can provide drivers with a customized system that behaves like their driving styles to promote acceptance and reliance.

- *Research Objective 2: Customized trust modeling for real-time prediction*

  This research objective aims to model and predict real-time human trust without interrupting and intruding. Trust is a hidden and implicit cognitive state. To integrate trust dynamics into the refinement of the customized systems proposed in the previous step, we need to first accurately model human trust while interacting with vehicle automation.

- *Research Objective 3: Trust-calibrated vehicle automation using human-in-the-loop method*

  This research objective aims to integrate the trust models into customized vehicle automation to generate trust-calibrated control policies. This objective takes a step toward human-aware automation that responds to human cognitive changes to induce improved reliance and acceptance.

The rest of the dissertation is organized as follows. In Chapter 2, we introduce our customized vehicle automation to driving styles using NDD. In Chapter 3, we present the customized trust models for real-time prediction. In Chapter 4, we demonstrate our method to integrate the trust models into the customized vehicle automation training process. Finally, in Chapter 5, we will discuss the conclusion, contribution, and potential future work.

Chapter 2

CUSTOMIZATION IN VEHICLE AUTOMATION

In this chapter, we introduce a proposed framework of customization in vehicle automation. The customized systems can capture drivers' preferences and differences by learning from their demonstrated driving behaviors and styles. This work is the fundamental part of the customized systems for later human-in-the-loop system design. This work has been accepted for publication in *IEEE Transactions on Vehicular Technology* [66].

2.1 Motivation and Objective

The field of vehicle automation is evolving rapidly. It seeks to enhance road safety and improve the driving experience. Although fully autonomous vehicles are not expected to be publicly available for several more years, most vehicles are equipped with some degree of vehicle automation to assist in the driving tasks. Vehicles with automation technologies promise significant societal benefits, including decreases in car crashes, injuries and deaths, enhanced mobility, increased road efficiency, and better utilization of parking and lands [69]. Even though research has shown substantial potential benefits of vehicle automation, the acceptance of these technologies is still not apparent, partly due to the significant increase in the variety of driving conditions and potential users.

Non-customized vehicle automation with a moderate driving style may be too conservative for more aggressive drivers and too aggressive for more passive drivers [22]. Some systems are customized to account for users' preferences using an in-vehicle interface [53, 110]. However, an effective interface design would require extensive usability studies with multiple real-world scenarios. Another approach is not based on user testing but rather on observing driving styles based on existing data to capture driver's behavior [58].

The objective of this study is to demonstrate the ability to customize lane change systems based on naturalistic data made available to the study team from the University of Michigan Transportation Research Institute (UMTRI). The system is targeted toward a SAE Level 2 system. The data provide insights on various driving styles for multiple drivers; this is an advantage over other models [22, 111] that focus only on one driver.

2.2 Related Work

Customizing a system based on the operator's capabilities and limitations is a data-driven approach. The premise of successful customization is identifying accurate and appropriate driving styles with representative driving behaviors using naturalistic driving data. In this section, we provide an extensive literature review of driving style identification. We then review the methods for customized automated lane change systems. The gaps observed in the literature are discussed in the context of the study motivation.

2.2.1 Driving Style Identification

Driving style is generally defined as the habitual ways drivers choose to drive [98]. It is crucial to accurately identify driving styles to help better design customized vehicle automation. However, driving style refers to all the activities performed by the driver, which contains many aspects. To date, a uniform method for quantifying or identifying driving styles has not been identified. However, qualitative definitions are available for some driving styles. This includes the actions that fall under the category of risky driving defined by NHTSA: drunk driving, distracted driving, speeding, frequent brakes, tailgating, etc. (https://www.nhtsa.gov/risky-driving). Some qualitative criteria can be transformed into quantitative measurements using continuous sensory data such as speed and distances to the surrounding vehicles. The speed profile provides information on the fluctuations and peaks related to aggressive or conservative driving styles [40]. The distance to the surrounding vehicles provides the behavior related to tailgating and abrupt lane changes. As long as these patterns are extracted in a reasonable manner, we can identify an effective driving style. A

survey by Martinez *et al.* [68] shows the typical process of recognizing meaningful driving styles. A summary of the driving style identification methods previously examined is shown in Fig. 2.1.

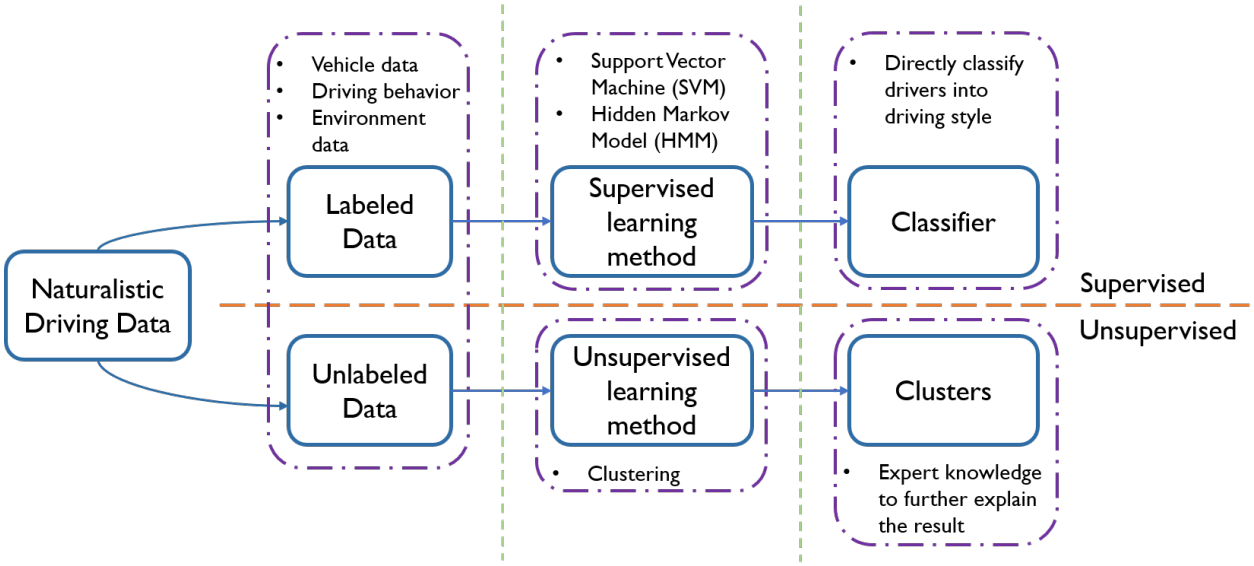


Figure 2.1: Two types of driving style identification methods in the recent literature.

Driving behaviors are typically captured using continuous sensory data from the vehicles. For example, speed profiles are collected over time and can, therefore, be examined using time series analysis methods [79, 106, 115, 117]. Speed profiles can also be examined using Functional Data Analysis (FDA). As compared to time series analysis methods, FDA does not require the data points to be evenly spaced and can model both temporal and spatial trends present in the data. In the past, FDA was attempted by [114] to examine the drivers' speed profiles in naturalistic driving data collected on rural two-lane curves. The results provide insights on driver behaviors for the individual road segments of different curvatures.

Once driving behaviors are successfully identified as constructed features, we can use supervised or unsupervised learning to distinguish the different driving styles. Supervised learning requires labeling the observations in the training dataset in advance [13, 63, 105, 112, 113]. Since naturalistic driving data is fairly noisy, the labeling process would be inaccurate

and biased by expert knowledge and personal judgments about the driving styles. The subjective labels can lead to underfitting or overfitting certain driving styles. Unsupervised learning, on the other hand, can assign each observation to a cluster based on the measured distances among the observations [19, 26, 48, 103]. Although it does not require extra effort to label the data, it might generate latent driving styles that were not previously known. For this reason, some manual identification or naming convention will be needed after the groups (or styles) are determined. Therefore, an ideal driving style identification method will need to include (1) automated feature extraction without complicated feature engineering and (2) selection using prior assumptions and interpretable clusters with minimum expert knowledge.

2.2.2 Customization in Automated Lane Change Systems

Personalization and customization have similar meanings and are sometimes used interchangeably. In this dissertation, customization refers to the adaptation of automation technologies to driving styles, whereas personalization captures the detailed individual preferences in driving tasks. Hence, we adopt the former term in our study but review the literature related to both in terms of vehicle automation.

The primary approach for designing a customized vehicle automation system is learning from demonstrations [58]. Here, the demonstrations refer to the operation data collected from the driving tasks. The widely adopted method is to observe human driving tasks and learn the behaviors using Inverse Reinforcement Learning (IRL) [124] or Imitation Learning [49,59]. Multiple studies adopted variants of these methods for personalized vehicle automation [25, 33, 84, 118].

For automated lane change systems, Butakov *et al.* [22] developed a method for modeling individual driver behaviors during lane changes. They used a sinusoidal lane change kinematic model and a Gaussian Mixture Model (GMM) to adjust the parameters to fit individual driving styles. However, the framework was validated on only three participants with 717 lane changes. The relatively high number of lane change behaviors for each participant

may be inefficient when applied to new users, and the limited number of participants can compromise the evaluation of the proposed method. In another study, Vallon *et al.* [111] proposed a data-driven classifier to predict the moment a lane change initiation occurs while also considering the preferences of the driver. This decision logic was then integrated into a model predictive control framework for lane change control. However, the limited number of participants impacts the validity of the model.

In another study, Zhu *et al.* [127] proposed a Personalized Lane-Change Warning System based on three components: surrounding vehicle information, the predicted safety distance based on the driving style, and a lane change feasibility judgment system. However, these modules were rule-based decision-making systems that did not learn or adapt to changes in the driver behaviors. Another recent work constructed a customized lane-change assistance system using deep learning and spatial-temporal modeling [40]. The results on real-world driving data showed that the lane-change model is capable of learning latent features and achieved better performance than the current system. However, stacking three sophisticated black-box algorithms weakened the interpretability of the model, and the reasoning process for the action became unclear to the user. In a very recent study, Huang *et al.* [53] identified the user's preferences using the fuzzy linguistic preference relation method, including measurements on driving safety, ride comfort, vehicle stability. However, the preferences need to be collected interactively with the passengers using a touch screen rather than passively identified from naturalistic driving data.

2.3 Methods

Customized automated lane change systems are built using a two-step method. The first step identifies the driving styles with representative behaviors from naturalistic driving data by clustering the driving trips into different groups. The second step customizes the automated lane change systems to the identified driving styles by applying the IRL method to the clustered trips (demonstrations). The IRL algorithm learns the latent preferences and objectives in the form of reward functions from the demonstrations. The reward functions

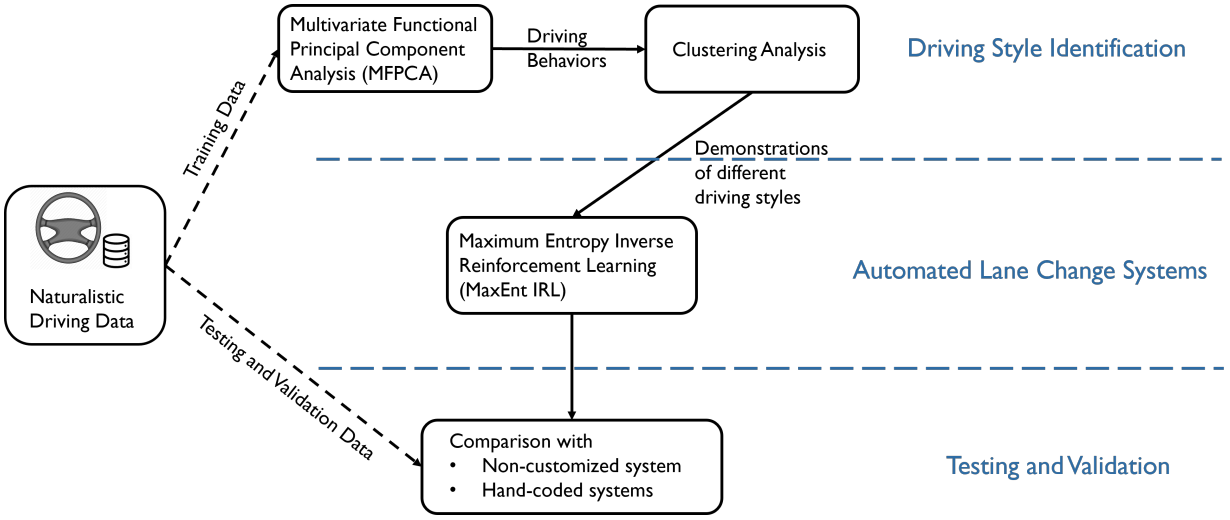


Figure 2.2: The overall framework of the proposed method.

encode the underlying intents in the lane change tasks, which ensures better performance than directly imitating the lane change behaviors. To achieve task-level decision making for lane changes using IRL, we need sequential lane change demonstrations for each driving style. Therefore, it is essential to identify the appropriate driving styles at the trip-level.

The overall framework consists of three components (see Fig. 2.2): a driving style identification method using Multivariate Functional Principal Component Analysis (MFPCA) and clustering analysis to generate grouped demonstrations for each driving style; automated lane change systems trained with an IRL method using the previously identified driving style demonstrations; and testing and validation of the proposed framework. We support the existence of the resulting driving styles with literature from behavioral psychology, and the interpretability of our driving style identification method ensures that the grouped demonstrations capture the representative behaviors of each driving style. In addition, the explicit forms of the reward functions extracted from the IRL method provide insights into the resulting behaviors of the customized automated lane change systems. The testing and validation highlight the improved action prediction performance of the proposed framework.

2.3.1 Driving Style Identification

The current gaps in driving style identification are addressed using MFPCA and clustering analysis. NDD is modeled as functional data to characterize the driver's behavior at the trip level. Such modeling can be used to identify the underlying patterns behind the driver behaviors and requires less prior knowledge and effort to calculate the characteristic features. These features are then used as inputs to identify the most relevant clusters for all the trips that lead to improved interpretability of the clustered driving styles. Finally, each cluster is labeled with an appropriate driving style based on similar behaviors.

Functional Data Analysis

The NDD are collected continuously over time. FDA is a collection of effective methods for analyzing functional data. It considers each individual datum as a function and does not impose any particular assumptions on the independence of the different values within a functional datum [91]. It, thereby, aims to detect the important patterns and explain the reasons behind the variations in the data [92]. These characteristics make FDA an appropriate method to extract features from naturalistic driving data.

It follows the same common notations used by the statistical models.

$$y = x(t) + \epsilon \tag{2.1}$$

where y is the response, $x(t)$ is the smooth function of a continuum t (usually time), and ϵ is the error term.

Specifically, we use the Basis Expansion model to represent the functional data:

$$x(t) = \sum_{i=1}^{K} c_j \phi_j(t) = \Phi(t)^T \mathbf{c} \tag{2.2}$$

where $x(t)$ is a single measurement in a continuum t, $\Phi(t)$ is a basis system, and $\mathbf{c}$ is the coefficient vector. Commonly-used basis systems include Fourier, B-spline, and Polynomial. However, in favor of stability, the former two basis systems are preferable in most applications. In practice, if the data is periodic, Fourier Basis is a better choice. Otherwise,

B-spline Basis is more suitable. B-splines are polynomial segments joined end-to-end which are constrained to be smooth at the joins. The points at which the segments join are called knots. The ith B-spline basis function of degree p (order $p+1$) is denoted by $N_{i,p}(t)$ and defined recursively as

$$
\begin{aligned}
N_{i,0}(t) &= \begin{cases} 1, & t_i \leq t < t_{t+1} \\ 0, & otherwise \end{cases} \\
N_{i,p}(t) &= \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t)
\end{aligned}
\tag{2.3}
$$

where t_i is the ith knot, $i = 0, 1, \ldots, n$, and $p \geq 1$.

A typical sum of squared error loss function is used to obtain the best fit coefficients. Since the smoothness is a key assumption in the functional data representation, a penalty term was added to induce smoothness. The general form of the loss function is represented in 2.4.

$$
PENSSE_{\lambda}(x) = \sum_{i=1}^{n} (y_i - x(t_i))^2 + \lambda J[x] \tag{2.4}
$$

where i is the index of the observations, λ is a continuous tuning parameter, and $J[x]$ is the smoothing penalty term. A common choice of the penalty term is $J[x] = \int (\frac{d^2}{dt^2} x(t))^2 dt$.

Multivariate Functional Principal Component Analysis

The typical basis expansion methods require us to define parameters, such as knots and orders, in advance. To address this issue, Functional Principal Component Analysis (FPCA) represents functional data in the most parsimonious way by using the eigenfunction basis to explain most of the variation. As the name suggests, FPCA is a variation of the traditional PCA in a functional basis system instead of a coordinate system.

The traditional multivariate PCA uses eigen-decomposition on the covariance matrix, but FPCA uses Karhunen-Loève decomposition. It finds the eigenvalue ρ_k and the eigenfunction

$\xi_k(t)$ by solving the eigenequation

$$\begin{aligned} \int \sigma(s,t)\xi_k(t)dt &= \rho_k\xi_k(s) \\ s.t. \int \xi_k(t)^2 dt &= 1 \\ \int \xi_k(t)\xi_l(t)dt &= 0, \quad l \neq k \end{aligned} \tag{2.5}$$

where $\sigma(s,t) = (N-1)^{-1}\sum_i x_i(s)x_i(t)$ is the covariance between curve values $x_i(s)$ and $x_i(t)$ at values s and t. Then corresponding score of a record r on ξ_k is $\int \xi_k(t)x_r(t)dt$.

An extension of univariate FPCA to the multivariate case is used to examine multiple dimensions of the sensory data (i.e., speed and relative locations of the surrounding vehicles). Happ and Greven [43] provide an algorithm to estimate the multivariate Principal Components (PCs) based on their univariate counterparts using the following steps.

1. Calculate the PC functions and scores using the Karhunen–Loève decomposition on the covariance surface.

2. Estimate the joint covariance matrix by combining all the PC scores into one large matrix.

3. Find the eigenvectors and eigenvalues of the estimated joint covariance matrix.

4. Calculate the estimated multivariate PC functions and scores by adjusting the univariate PC functions and scores from step 1 using the eigenvectors and eigenvalues estimated in step 3.

Clustering Analysis

A k-means clustering algorithm is applied to the scores of the functional PCs. The k-means algorithm computes the centroid of each cluster iteratively and updates the assignment of each observation. While converging to stable assignments, k-means finds the clusters that minimize the within-cluster variances. There have been previous attempts to implement

k-means clustering on functional PCs scores, and results confirm the effectiveness of the method [39]. Driving style identification is achieved once each observation is assigned to an appropriate cluster.

Algorithm 1: K-means clustering

Input : Observations $x^{(i)}$

Output: Assigned clusters for every observation $c^{(i)}$

Initialize cluster centroids $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}$ randomly;

while *not converge* **do**

  for *every* i **do**

    $c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2$

  for *every* j **do**

    $\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)}=j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)}=j\}}$

2.3.2 Customized Vehicle Automation

The driving style identification step clusters the driving trips into different driving styles. In the second step, we leverage the grouped driving demonstrations generated by the previous step and adapt a data-driven method to customize the automated lane change systems. Each customized system is trained separately from the driving demonstrations.

Reinforcement Learning

The vehicle automation system can be regarded as an intelligent agent that can be trained using Reinforcement Learning (RL). RL provides the learning ability to make sequential decisions in order to improve the agent's experience through interactions with its environment instead of explicit programming. This is achieved by maximizing the expected cumulative reward of the agent's actions, where the environment is typically modeled as a Markov Decision Process (MDP). An MDP is defined by four components: a set of states S, a set of actions A, a reward R, and a state transition probability P. The state is a collection of

variables that characterize the system (agent and its environment) at any point in time, and the action is the decision made by the agent to transition from one state to the next to maximize the cumulative reward over a given time horizon. Here, the state is characterized by the variables that affect lane change decisions such as the speed of the ego car and its distance to the surrounding vehicles. Several other important definitions and properties of an MDP:

- $R_a(s, s')$ is the immediate reward received after transitioning from state s to state s' by taking action a.
- $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ is the probability of transitioning to state s' at time $t + 1$ from state s by taking action a at time t.
- $\pi(a_t|s_t)$ is the probability of taking action a_t in state s_t at time t. $\pi(s) \in A$ is the optimal action in state s at time t.
- γ is the discount factor.
- Markov Property: $P(S_{t+1}|S_t) = P(S_{t+1}|S_1, S_2, \ldots, S_t)$
- $V_\pi(s) = E[R(s_t) + \gamma V_\pi(s_{t+1})|S_t = s]$ is the value function.
- $Q_\pi(s, a) = E[R(s_t) + \gamma Q_\pi(s_{t+1}, a_{t+1})|S_t = s, A_t = a]$ is the action-value function.

The goal of RL is to find an optimal policy that maximizes the expected future reward.

$$\arg\max_\pi E[\sum_{t=0}^{T} R_{\pi(s_t)}(s_t, s_{t+1})] \tag{2.6}$$

where the expectation is taken over $s_{t+1} \sim P_{\pi(s_t)}(s_t, s_{t+1})$.

There are a lot of RL algorithms that can solve the above optimization problem efficiently. Depending on whether the transition probability $P_a(s, s')$ is known, they can be categorized into Model-free RL and Model-based RL. For the first kind, if the algorithm is to learn

the policy directly, some examples are Asynchronous Advantage Actor-Critic (A3C) [73], Proximal Policy Optimization (PPO) [100]. Otherwise the algorithm is to learn the action-value functions, then the examples are Deep Q-Network (DQN) [74], Hindsight Experience Replay (HER) [12]. For the second kind, the examples are Model-Based RL with Model-Free Fine-Tuning (MBMF) [82], and Model-Based Value Expansion (MBVE) [37].

Maximum Entropy Inverse Reinforcement Learning

RL has been shown to be immensely successful in a wide variety of applications. However, unlike most other applications, no specific reward signals are directly generated from the driving tasks in our study. We therefore used an IRL algorithm to extract the specific form of the reward function.

The concept of IRL is first introduced by Ng and Russel in 2000 [86]. They formally define some important notations of IRL and state the motivation behind IRL. They propose three algorithms for solving simple IRL problems. Building upon their own work, they propose Apprenticeship Learning to solve IRL problems [1]. They propose to use linear function approximation to estimate the reward function which is linear to the state features. This assumption can lead to sufficient solutions to the IRL algorithm. However, as they discussed in the paper, one of the most challenging issues in the IRL algorithm is the algorithm will find a solution set with even degenerate one instead of a single optimal reward function. All the reward functions in the solution set will satisfy all the constraints and potentially be able to generate the observed demonstrations or optimal policy. It's difficult to choose the optimal reward function without any premises.

As the topic becomes more and more popular, a lot of variations of IRL algorithms are proposed. Most of them are trying to address the above-mentioned issue. The most influential method is built on the Principle of Maximum Entropy [55]. The entropy of a continuous random variable X is defined as:

$$H(X) = -\int_{-\infty}^{\infty} p(x) \log p(x) dx \tag{2.7}$$

For discrete random variable, the integral is replaced by the summation.

Ziebart et al. [128] propose Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) to resolve ambiguities in choosing distributions. The maximum entropy principle leads to the choice of optimal reward function with the maximum entropy. There are a few definitions introduced to complement the notations of MDP before introducing the MaxEnt IRL algorithm:

- $\zeta = \{(s_1, a_1), (s_2, a_2), (s_3, a_3), \dots\}$ is a demonstrated trajectory.
- $f_s \in \mathbb{R}^k$ is the feature vector of the state s.
- $f_\zeta = \sum_{s \in \zeta} f_s$ is called the feature counts.
- $\theta \in \mathbb{R}^k$ is the reward function parameters.
- $P(\zeta)$ is the probability of observing the trajectory ζ.
- $P(s)$ is the probability of visiting state s, also called state visitation frequency.

A direct result coming from the additional notations is:

$$P(\zeta) = \prod_{s \in \zeta} P(s) \tag{2.8}$$

They assume the reward of a trajectory is linear to the feature counts.

$$R(\zeta) = \theta^T f_\zeta = \sum_{s \in \zeta} \theta^T f_s \tag{2.9}$$

The probability of a trajectory demonstrated by the expert (in our case, the human driver) has to be exponentially greater for larger rewards as compared to smaller rewards. Mathematically,

$$P(\zeta) \propto e^{R(\zeta)} \tag{2.10}$$

The MaxEnt IRL algorithm solves for optimal θ^* by maximizing the likelihood of the demonstrated trajectories,

$$\begin{aligned}\theta^* &= \arg\max_\theta L(\theta) \\ &= \arg\max_\theta \frac{1}{M} \log \prod_{\zeta_D} P(\zeta_D|\theta) \\ &= \arg\max_\theta \frac{1}{M} \sum_{\zeta_D} \log P(\zeta_D|\theta) \\ &= \arg\max_\theta \frac{1}{M} \sum_{\zeta_D} R(\zeta_D) - \log \sum_{\zeta} e^{R(\zeta)}\end{aligned} \tag{2.11}$$

where M is the number of trajectories in demonstration, ζ_D is the observed trajectory from demonstration, and ζ is any trajectory that can be observed from the task.

The above objective function is convex , so the optimal solution can be solved using a gradient descent based method. The gradient is,

$$\begin{aligned}\nabla_\theta L(\theta) &= \frac{1}{M} \sum_{\zeta_D} f_{\zeta_D} - \frac{1}{\sum_\zeta e^{R(\zeta)}} \sum_{\zeta} (e^{R(\zeta)} \frac{dR(\zeta)}{d\theta}) \\ &= \frac{1}{M} \sum_{\zeta_D} f_{\zeta_D} - \sum_{\zeta} \frac{e^{R(\zeta)}}{\sum_\zeta e^{R(\zeta)}} f_\zeta \\ &= \frac{1}{M} \sum_{\zeta_D} f_{\zeta_D} - \sum_{\zeta} P(\zeta|\theta) f_\zeta \\ &= \frac{1}{M} \sum_{\zeta_D} f_{\zeta_D} - \sum_{s} P(s|\theta) f_s\end{aligned} \tag{2.12}$$

where $P(s|\theta)$ is the state visitation frequency given the reward parameter θ. Define $\mu_t(s|\theta)$ as the probability of visiting state s at time t. It can be calculated efficiently using Alg. 3. With the state visitation frequency calculated and the value iteration shown in Alg. 2, the MaxEnt IRL algorithm is shown in Alg. 4.

Algorithm 2: Value Iteration

for $k = 1, 2, \ldots, \infty$ **do**
  for *each state* s **do**
    $V_k(s) = \max_a \sum_{s'} P_a(s, s')(R(s) + \gamma V_{k-1}(s'))$
  if $\forall s, |V_k(s) - V_{k-1}(s)| < threshold$ **then**
    for *each state* s **do**
      $\pi(s) = \arg\max_a \sum_{s'} P_a(s, s')(R(s) + \gamma V_{k-1}(s'))$ **return** π, V_k
  else
    continue

Algorithm 3: State Visitation Frequency Calculation

Solve the MDP using value iteration for optimal policy π;

Computer $\mu_1(s)$ using sampled trajectories;

for $t = 1, 2, \ldots, T$ **do**
  $\mu_{t+1}(s|\theta) = \sum_a \sum_{s'} \mu_t(s')\pi(a|s')P_a(s, s')$
return $P(s|\theta) = \sum_t \mu_t(s|\theta)$

Algorithm 4: MaxEnt IRL algorithm

0. Initialize θ randomly;
1. Solve for optimal policy π w.r.t θ with value iteration;
2. Solve for state visitation frequency $P(s|\theta)$ using Algorithm 3;
3. Compute gradient $\nabla_\theta L(\theta) = \frac{1}{M} \sum_{\zeta_D} f_{\zeta_D} - \sum_s P(s|\theta) f_s$;
4. Update θ with one gradient step using $\nabla_\theta L(\theta)$

The MaxEnt IRL algorithm demonstrates great performance in various applications. It can handle expert suboptimality and stochasticity by operating on the distribution over possible trajectories. However, the linear function approximation of the reward function is sometimes too rough. For complicated tasks, the reward function is likely to be nonlinear in the feature space. To better approximate the nonlinear structure of the reward function, Neural Networks is proposed to replace the linear function approximation assumption in the

maximum entropy framework, which is named Maximum Entropy Deep Inverse Reinforcement Learning (DeepIRL) [121].

As a generalization of MaxEnt IRL, DeepIRL uses Deep Neutral Networks to map the state features to reward values. It is straight forward to show that the MaxEnt IRL is a special case of DeepIRL by applying back-propagation to a network with a single linear output connected to all inputs with zero bias term. Suppose the Neural Network approximation of the reward function can fit in the following general framework:

$$R(\zeta) = \sum_{s \in \zeta} R(s) = \sum_{s \in \zeta} g(f_s, \theta) \tag{2.13}$$

Under the same derivation procedure, MaxEnt IRL and DeepIRL share the same maximum likelihood objective function. The difference is in the derivation of Eq. 2.12.

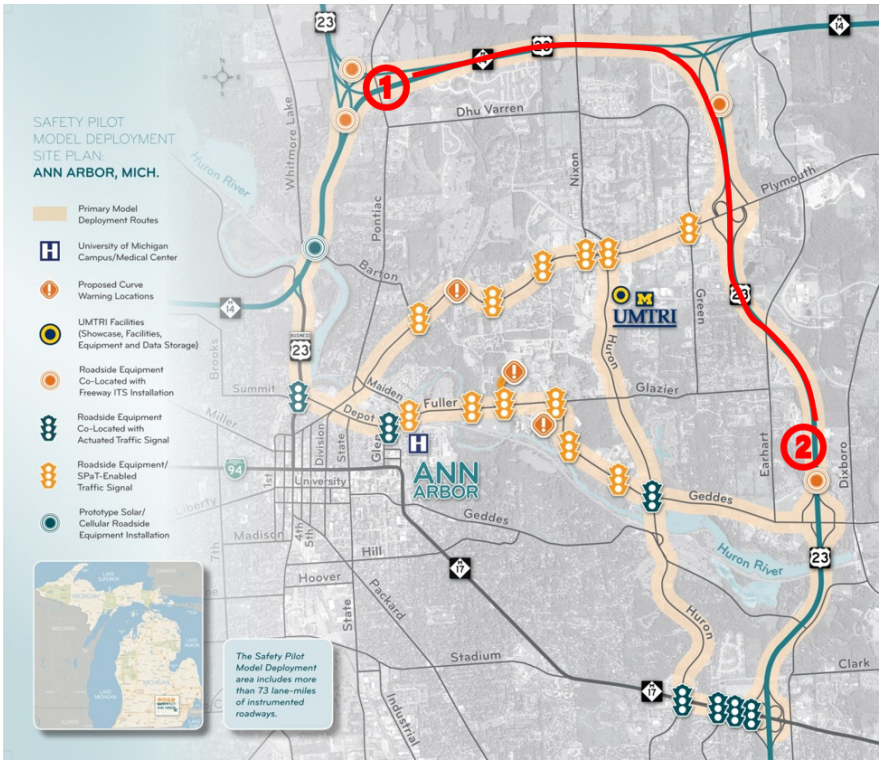
$$\nabla_\theta L(\theta) = \frac{1}{M} \sum_{\zeta_D} \frac{\partial}{\partial \theta} R(\zeta) - \frac{1}{\sum_\zeta e^{R(\zeta)}} \sum_\zeta (e^{R(\zeta)} \frac{\partial}{\partial \theta} R(\zeta)) \tag{2.14}$$

where $\frac{\partial}{\partial \theta} R(\zeta)$ can be computed efficiently using back-propagation for Neural Networks. The rest of the algorithm is exactly the same as MaxEnt IRL.

2.4 Naturalistic Driving Data

The dataset used for this study comes from the UMTRI Safety Pilot Model Deployment (SPMD) naturalistic driving data [20]. The data collection period was from August 2012 to June 2017. This dataset is a part of the Connected Vehicle Safety Pilot Program, a research initiative that features real-world implementation of connected vehicle safety technologies, applications, and systems using everyday drivers. SPMD data includes approximately 3000 vehicles (some with more than one driver) equipped with vehicle-to-vehicle (V2V) communication in real-world conditions at a test site with diverse traffic [20].

Fig. 2.3 shows the layout of the test site and the location of the roadside equipment used in data collection. There were five data acquisition systems that worked cooperatively to record four sets of data: contextual (e.g., weather, lighting), driving (from a vehicle's data



Note: Red circles denote the approximate starting (1) and ending (2) locations of the trips for the highway segment of interest (red solid line).

Figure 2.3: Safety Pilot Model Deployment Site Plan in Ann Arbor, Michigan. Modified from [20].

acquisition system (DAS)), basic safety messages (BSM), and data from roadside equipment (RSE). The SPMD data contains more than four terabytes of raw data. We used the publicly available data, which covered a 7-month period from October 2012 to April 2013 (`https://catalog.data.gov/sv/dataset/safety-pilot-model-deployment-data`). This dataset does not include any personal identifiers and is exempt from additional Institutional Review Boards (IRB) review. The original IRB process is described in [20].

The road environment varied from low-speed urban roads to high-speed freeways. For our analysis, a segment of the freeway driving route was selected. It starts on location 1 and

ends on location 2 (shown in red in Fig. 2.3). The trips in the counterclockwise direction were also selected for the same freeway segment, which starts on location 2 and ends on location 1. The selected route is approximately 7 kilometers with 3.5 minutes of travel time. Note that the selected segment has a junction from U.S. Route 23 to U.S. Route 14 in the elbow transition part. Specifically, we filtered the trips based on the GPS coordinates from the DAS to select the road segments of interest. We then extracted four variables for analysis: (1) speed of the ego car, (2) distance to the vehicle in front, and (3) average distances to the vehicles on the left and (4) the right. Previous studies have shown that these features reflect the driving behaviors and significantly affect the lane change decisions [7, 40]. As a result, a total of 105 trips, including 46 in the clockwise direction and 59 in the counterclockwise direction, were selected for further analysis.

2.5 Results

The experiments were carried out on a MAC Big Sur version 11.6 Operating System using R programming language version 3.6.1 and Python version 2.7. A 2.6 GHz Intel Core i7 quad-core processor with 16 GB 2133 MHz LPDDR3 RAM was used. R packages mfpca [44] and other supportive packages were employed for implementing MFPCA. Python packages Theano [107] and numpy [45] were used for implementing IRL algorithm.

2.5.1 Driving Style Identification

Four variables from the naturalistic driving data were selected to identify the driving styles. The speed profiles provided data on speeding, and changes in speed over time (i.e., acceleration/deceleration). The distances to the surrounding vehicles were used to identify aggressive driving behaviors (e.g., tailgating and abrupt lane changes). We then applied the MFPCA method to extract the driving behaviors.

For the clockwise direction, the top four functional principal components are able to explain 92.3% of the total variability. The distances to the surrounding vehicles were missing in many observations. Hence, most of the variability came from the speed profile, and only a

small amount was from the distance to the vehicles in front. The functional PC 2 is shown in Fig. 2.4. Additional MFPCA results are presented in Appx. A.1 The corresponding coefficient of the average speed profile for this functional PC is 32.2. Therefore, a coefficient greater than 32.2 indicates that the trip has a lower than average speed and faster deceleration than average trips; otherwise, the trip has the opposite behavior. Similarly, other driving behaviors can be revealed for the other functional PCs:

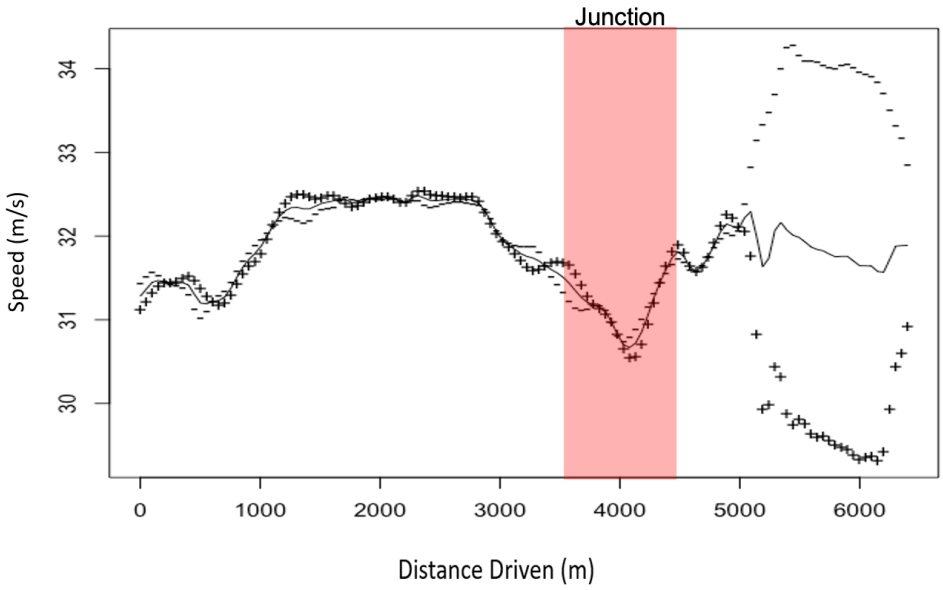
1. Functional PC 1: Variability is observed before entering the junction segment. A coefficient larger than the average speed profile for FPC1 indicates that the driver tends to have a greater than average speed and faster than average deceleration before entering the junction.

2. Functional PC 2: Variability is observed at the long open road after the junction. A coefficient larger than the average speed profile for FPC2 (in this case 32.2) corresponds to lower than average speed and larger than average distance to the vehicle in front.

3. Functional PC 3: Variability is spotted when entering the freeway. A coefficient larger than the average speed profile for FPC3 means the driver has a slower than average speed and prefers larger than average distance to the vehicle in front.

4. Functional PC 4: Variability is observed before entering the junction segment. A coefficient larger than the average speed profile for FPC4 denotes greater speeding behavior on the open road before entering the junction segment.

We also applied k-means clustering on the extracted MFPCA scores. The silhouette analysis showed that three clusters provided the largest average silhouette coefficient (0.56) among clusters varying from two to six. This is consistent with studies from traffic psychology [8, 116] that show three broad categories of driving styles: aggressive, neutral, and conservative.



Note: The solid line denotes the average driving profile of all the trips (coefficient=32.2). A coefficient greater than 32.2 will move the speed profile closer to the "+" line; otherwise, the profile is closer to the "-" line. More specifically, the "+" and "-" lines are generated by adding or subtracting two standard deviation of the functional PC from the average speed profile, respectively.

Figure 2.4: The effect of functional PC 2 on the average speed profile. FPC 2 explains 21.1% of the variability in the clockwise direction.

We visualized the three resulting clusters using the speed profile in Fig. 2.5 since it encodes the most variability in the data. The clusters are labeled as aggressive trips, neutral trips, and conservative trips. The aggressive trips tend to have higher speeds, faster accelerations, and a small distance to the front vehicles on the open road after and in the junction segments. The conservative trips demonstrate the opposite behaviors, and the neutral trips lie between the two extremes.

The same analysis steps were conducted for the counterclockwise direction. The top four functional PCs explained 86.1% of the total variability. The same issue with missing values

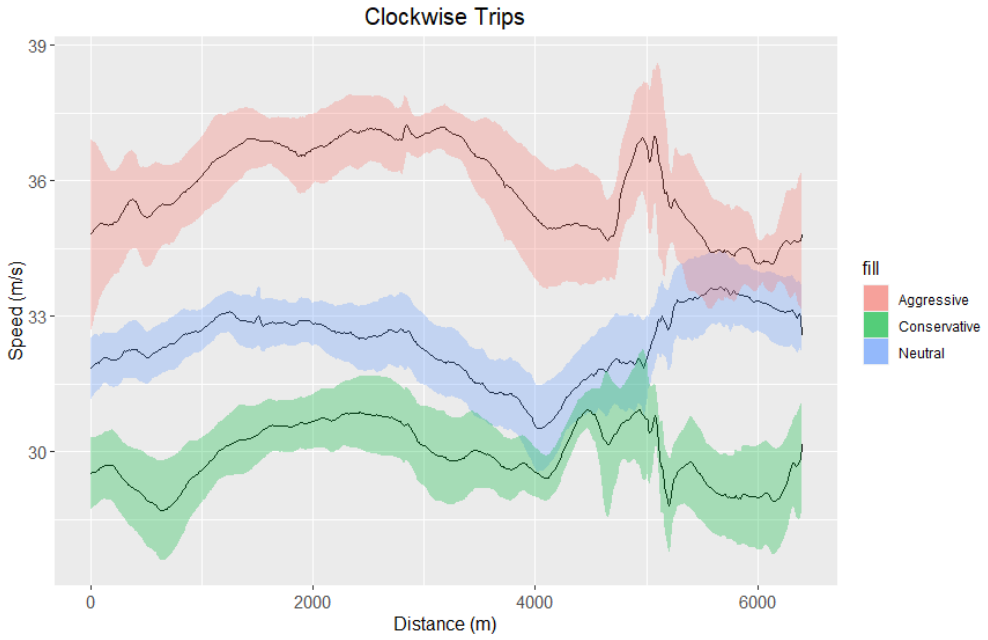


Figure 2.5: Three clusters for the clockwise direction: each cluster is depicted by its mean speed and the corresponding 95% confidence interval.

in the distances to the surrounding vehicles was observed. The top four FPCs represented the following behaviors:

1. Functional PC 1: Variability is observed when leaving the highway. A coefficient larger than the average speed profile for FPC1 indicates that the driver tends to have a relatively high speed when leaving the highway.

2. Functional PC 2: Variability is observed before the junction segment. A coefficient larger than the average speed profile for FPC2 denotes fast speed, rapid deceleration, and small distance to the vehicle in front.

3. Functional PC 3: Variability is spotted on the long open road after the junction segment. A coefficient larger than the average speed profile for FPC3 indicates speeding behavior in the long open road after the junction.

4. Functional PC 4: Variability is spotted in the open road after the junction. A coefficient larger than the average speed profile for FPC4 indicates that the driver is braking first after the junction, but accelerating fast thereafter. Otherwise, the driver accelerates fast at first, but then slows down.

Fig. 2.6 summarizes the speed profiles of the three clusters identified by the k-means clustering method for the counterclockwise direction trips. Since the conservative group contains a small number of trips, its confidence intervals are wider than the other two groups.

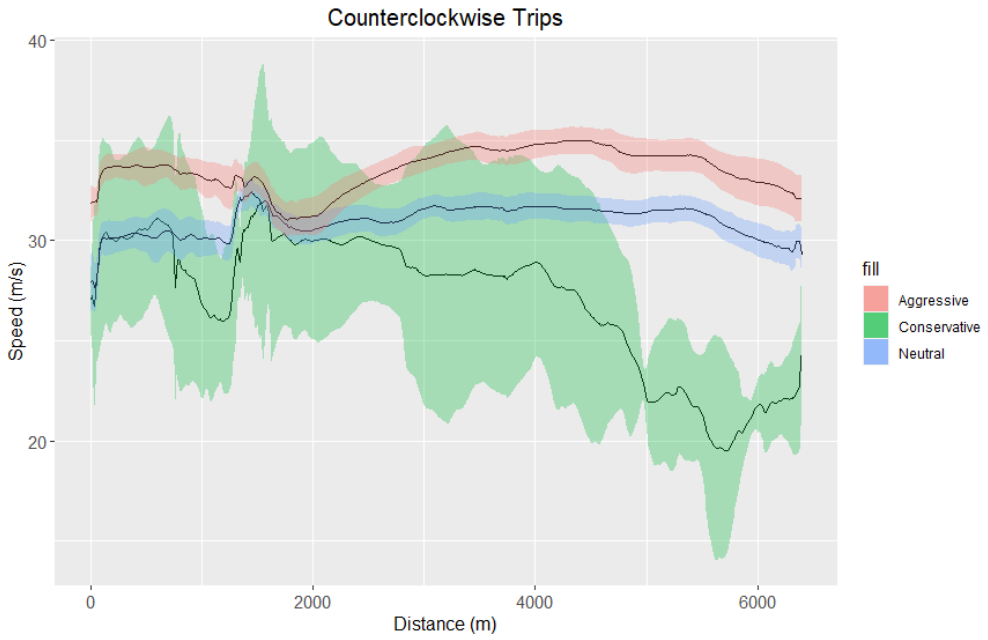


Figure 2.6: Three clusters for the counterclockwise direction: each cluster is depicted by its mean speed and the corresponding 95% confidence interval.

In summary, out of 105 total trips, 27 are classified as aggressive, 56 as neutral, and 22 as conservative. These behaviors, as revealed by the spatial and temporal characteristics of the driving trips using MFPCA, match the qualitative description of the corresponding driving styles discussed in Section 2.3.1. Previous studies have also shown that these factors

significantly impact lane change decision making [7, 77]. Therefore, the results confirm that our driving style identification method is able to extract representative driver behaviors, and clustering analysis successfully separates the different driving styles with minimal feature engineering and prior knowledge about the driving styles.

2.5.2 Customized Vehicle Automation

We customized the automated lane change systems using the IRL method, wherein three separate systems were trained on the clusters generated by the driving style identification method. Specifically, three reward functions were learned from the clustered demonstrations to capture the preferences of the different driving styles. Accordingly, the lane change systems followed the optimal (stochastic) policies corresponding to the reward functions to adapt or customize to the different driving styles.

The lane change system is considered one of the most challenging vehicle automation problems, since it involves both longitudinal and lateral vehicle control, and requires constant monitoring of the surroundings. For this reason, we selected the same four features for the IRL method: the speed of the ego vehicle, the distance to the lead vehicle, the average distance to the right vehicles, and the average distance to the left vehicles.

The IRL algorithm is computationally expensive in recovering reward functions and stochastic policies. The surrounding vehicle information has a high percentage of missing values due to noisy and imperfect naturalistic driving data. Therefore, we discretize the continuous state space into a discrete state space. Each continuous variable is discretized into four levels based on the first, second, and third quantile values along with an additional level to encode either missing information or no car in a given direction. The detailed numerical intervals for each level are provided in Tab. A.1 of Appx. A.2. As a result, the state space contains 500 ($4 \times 5 \times 5 \times 5$) distinct states and 19 ($4 + 5 + 5 + 5$) feature variables (see Tab. 2.1). The action space comprises three actions: staying in the current lane, changing to the right lane, or changing to the left lane.

The transition probability is obtained by taking the maximum likelihood estimate of

Table 2.1: Discretization of the continuous variables: speed, forward distance, left distance, and right distance.

Var. Names	Speed	Forward Dist.	Left Dist.	Right Dist.
List of Levels	Slow	No Car	No Info	No Info
	Medium	Very Small	Very Small	Very Small
	Fast	Small	Small	Small
	Very Fast	Medium	Medium	Medium
		Large	Large	Large

Note: The 'No Info' level corresponds to a situation with no car or no lane information.

the transition matrix [102]. However, not all states are guaranteed to be observed in the demonstrated trajectories. To account states with no observations, we adopt the common ϵ-greedy ($\epsilon = 0.1$) algorithm to get a trade-off between exploration and exploitation. The experiments are carried out using a Feed-forward Neural Network (FNN) with two hidden layers of 30 hidden units in each layer. We select the discount factor as 0.9, and the learning rate as 0.01. The reward values are normalized to a continuous scale between -1 and 1. The results are presented for the recovered reward functions and stochastic policies. The optimal policies are recovered using approximate value iteration in each update of the reward function.

Summary statistics of the reward functions for four representative driving scenarios are shown in Tab. 2.2. We selected these scenarios to demonstrate the interpretability and effectiveness of the IRL method. In the first three scenarios, we fix the ego vehicle speed and the distance to the forward vehicle, and show the summary statistics as a result of varying the other two variables (average distance to the left vehicles and right vehicles). When the ego vehicle is slow and there is no lead vehicle, the aggressive drivers have the least number of non-negative reward values and the conservative drivers have the most number of such

Table 2.2: Summary statistics of the reward values in four example scenarios.

Scenarios (variable: level)	Driving Style	Non-negative portion*	Min	Median	Mean	Max	SD
Speed: Slow Forward Dist.: No Car	Aggressive	12/25	-0.418	-0.014	0.045	**1**	0.275
	Neutral	15/25	**-0.059**	0.074	0.048	0.398	0.255
	Conservative	**21/25**	-0.114	**0.210**	**0.285**	**1**	**0.318**
Speed: Fast Forward Dist.: Very Small	Aggressive	**8/25**	**-0.101**	**-0.037**	**-0.004**	**0.403**	0.111
	Neutral	7/25	-0.269	-0.046	-0.048	0.158	0.080
	Conservative	5/25	-0.343	-0.174	-0.120	0.293	**0.172**
Speed: Medium Forward Dist.: Large	Aggressive	11/25	**-0.087**	-0.018	0.044	**0.520**	0.165
	Neutral	**18/25**	-0.279	**0.069**	**0.065**	0.357	0.128
	Conservative	10/25	-0.489	-0.079	-0.077	0.320	**0.203**
Left Dist.: Medium Right Dist: Medium	Aggressive	7/16	-0.137	**-0.048**	-0.001	0.281	0.114
	Neutral	**8/16**	**-0.108**	**-0.048**	**0.009**	**0.326**	0.109
	Conservative	3/16	-0.324	-0.146	-0.129	0.160	**0.118**

Note: *Non-negative values indicate that the driver is more likely to stay in the scenario.
Bold represents the largest value for a particular scenario and driving style.

values. The reward values for the neutral drivers lie in between those for the aggressive and conservative drivers. Moreover, the conservative drivers have the largest minimum, median, and mean reward values, which means that they are more comfortable staying in this scenario as compared to the other two driving groups.

On the other hand, the scenario with fast ego vehicle speed and very small forward distance is preferred by the aggressive driver group. Likewise, the scenario with the medium ego vehicle speed and large forward distance is preferred by the neutral driver group. We also fixed the average distance to the right vehicles and left vehicles as medium, and examined the reward values of the other two variables. In this scenario, the neutral drivers have the largest minimum, median, mean, and maximum reward values. Thus, we observe that the optimal policies obtained by the learned reward function facilitate model interpretability.

The recovered policies are used to validate the proposed IRL method by comparing the behaviors associated with the different driving styles. Tables 2.3, 2.4, 2.5 present the stochastic policy calculated for aggressive drivers, neutral drivers, and conservative drivers based on the recovered reward functions, respectively. The results show that the aggressive drivers prefer to switch lanes in more than half of the scenarios. The optimal actions would still maximize safety by choosing actions with the highest probabilities. For example, although the probability of changing to left or right lanes in the small distance scenarios is 0.286 for aggressive drivers (see Tab. 2.3), the optimal action is to stay in the current lane since it has the highest probability of 0.429. Alternatively, the conservative drivers are quite comfortable traveling at a low speed in the current lane even when there is no lead vehicle. Finally, the neutral drivers prefer to stay in the current lane for most scenarios, but are willing to switch to other lanes in a few cases. To better demonstrate the performance of the recovered optimal policies, additional policy tables are provided for a second scenario of fast driving speed and a very small distance to the lead vehicle in Appx. A.2.

2.5.3 Performance Evaluation

The performance of the proposed customized automated lane change system is compared to (1) a non-customized system trained on all the sample trips, (2) customized systems built on expert-coded reward functions, and (3) customized systems trained using a Generative Adversarial Imitation Learning (GAIL) algorithm.

The non-customized system is expected to have an average driving style across all the sample trips. The optimal policy is trained using the same IRL method on all the trips. Specifically, we design three expert-coded reward functions, one for each driving style group, using an approach similar to An *et al.* [11] and Hoel et al. [50]. Their primary approach is to balance the efficiency and safety of automated lane change systems. To maximize safety, the reward function is purposely coded to give a large negative reward if the vehicle lands in a collision or near-collision state. We define the near-collision state as being very close to the surrounding vehicles and use similar reward values (i.e., -5, -10, and -20 for aggressive,

Table 2.3: Stochastic policy of the aggressive drivers for slow driving without any lead car.

Dist. to Left Vehicles	Dist. to Right Vehicles	Probability of Action		
		Stay in Current Lane	Change to Right Lane	Change to Left Lane
No info	No info	0.476	0.313	0.211
No info	Very small	0.152	0.424	0.424
No info	Small	0.323	0.292	0.385
No info	Medium	0.26	0.288	0.452
No info	Large	0.266	0.718	0.016
Very small	No info	0.634	0.221	0.144
Very small	Very small	0.429	0.286	0.286
Very small	Small	0.296	0.352	0.352
Very small	Medium	0.355	0.171	0.474
Very small	Large	0.133	0.433	0.433
Small	No info	0.273	0.63	0.097
Small	Very small	0.499	0	0.501
Small	Small	0.47	0.265	0.265
Small	Medium	0.346	0.327	0.327
Small	Large	0.36	0.269	0.371
Medium	No info	0.34	0.268	0.391
Medium	Very small	0.364	0.318	0.318
Medium	Small	0.843	0.078	0.078
Medium	Medium	0.571	0.215	0.215
Medium	Large	0.42	0.29	0.29
Large	No info	0.522	0.2	0.278
Large	Very small	0.285	0.357	0.357
Large	Small	0.38	0.42	0.2
Large	Medium	0.491	0.255	0.255
Large	Large	0.119	0.055	0.826

Note: Blue shade denotes the optimal action

neutral, and conservative drivers, respectively). An additional negative reward is designated if the system tries to choose an action that results in a near-collision state (i.e., -1.5, -3, and -6, respectively). In terms of efficiency, a positive reward is given such that it is proportional to either the driving distance or the average speed between two consecutive actions. Since we discretize speed into 4 levels and the data is recorded at 10 Hz, we assign the same positive reward for any particular speed level in each of three driving styles, i.e., 1, 3, 5 and 7 for slow, medium, fast and very fast speed, respectively.

Table 2.4: Stochastic policy of the neutral drivers for slow driving without any lead car.

Dist. to Left Vehicles	Dist. to Right Vehicles	Probability of Action		
		Stay in Current Lane	Change to Right Lane	Change to Left Lane
No info	No info	0.546	0.407	0.047
No info	Very small	0.406	0.332	0.262
No info	Small	0.398	0.172	0.43
No info	Medium	0.527	0.267	0.206
No info	Large	0.432	0.363	0.206
Very small	No info	0.266	0.389	0.345
Very small	Very small	0.346	0.252	0.402
Very small	Small	0.509	0.436	0.054
Very small	Medium	0.147	0.62	0.233
Very small	Large	0.322	0.576	0.102
Small	No info	0.683	0.154	0.162
Small	Very small	0.367	0.42	0.213
Small	Small	0.424	0.288	0.288
Small	Medium	0.39	0.343	0.267
Small	Large	0.378	0.444	0.177
Medium	No info	0.57	0.43	0
Medium	Very small	0.662	0.168	0.171
Medium	Small	0.493	0.253	0.253
Medium	Medium	0.696	0.017	0.287
Medium	Large	0.718	0.141	0.141
Large	No info	0.452	0.407	0.141
Large	Very small	0.279	0.23	0.491
Large	Small	0.549	0.221	0.23
Large	Medium	0.718	0.141	0.141
Large	Large	0.783	0.109	0.109

Note: Blue shade denotes the optimal action

Compared to the learned reward function using IRL, the expert-coded reward depends much less on the state levels. It assigns the same reward value for every near-collision state. However, the learned reward function distinguishes between the different near-collision states. For instance, if the distance from the vehicle on the right is very small, but the distance from the vehicle on the left is large, the reward value is positive, whereas the expert-coded reward function generates negative values.

Lastly, we implemented a GAIL algorithm to learn the optimal policies from the demon-

Table 2.5: Stochastic policy of the conservative drivers for slow driving without any lead car.

Dist. to Left Vehicles	Dist. to Right Vehicles	Probability of Action		
		Stay in Current Lane	Change to Right Lane	Change to Left Lane
No info	No info	0.63	0.326	0.044
No info	Very small	0.425	0.287	0.287
No info	Small	0.359	0.321	0.321
No info	Medium	0.521	0.246	0.233
No info	Large	0.544	0.228	0.228
Very small	No info	0.951	0.025	0.025
Very small	Very small	0.691	0.155	0.155
Very small	Small	0.506	0.247	0.247
Very small	Medium	0.397	0.302	0.302
Very small	Large	0.501	0.25	0.25
Small	No info	0.973	0.014	0.014
Small	Very small	0.592	0.131	0.277
Small	Small	0.753	0.124	0.124
Small	Medium	0.709	0.146	0.146
Small	Large	0.758	0.121	0.121
Medium	No info	0.715	0.199	0.086
Medium	Very small	0.415	0.2	0.385
Medium	Small	0.403	0.299	0.299
Medium	Medium	0.452	0.274	0.274
Medium	Large	0.695	0.153	0.153
Large	No info	0.917	0.042	0.042
Large	Very small	0.935	0.032	0.032
Large	Small	0.576	0.212	0.212
Large	Medium	0.677	0.162	0.162
Large	Large	0.888	0.056	0.056

Note: Blue shade denotes the optimal action

strated driving styles. The GAIL algorithm used in this study was adopted by Ho and Ermon [49], and was shown to be equivalent to the MaxEnt IRL algorithm for a specific form of the discriminator loss function [38]. However, this GAIL algorithm is a form of imitation learning that learns the optimal policies directly from the demonstrations without any specific form of the reward functions.

For brevity, we only present the stochastic policy of the conservative drivers in Tab. 2.6, where we fix the ego car speed to be slow and do not have any lead vehicle. A significant

difference is observed between the optimal policies for the expert-coded and IRL methods. Tab. 2.5 suggests that the optimal policy for the IRL method is to always stay in the current lane. However, the optimal policy for the expert-coded reward function shows some unexpected behaviors. For example, it tries to change to the right lane even though there is already a very small distance to the vehicles on the right. Additionally, there are several states that have an equal probability of changing to the right and left lanes. It suggests the agent is not learning useful information from the expert-coded reward function. Lastly, most probabilities in the table are 0, 0.5, or 1, which means the optimal policy is not able to capture the stochasticity in the actions.

The policy tables for the aggressive and neutral drivers are provided in Appx. A.2. For the aggressive policy recovered from the expert-coded reward function, there are more lane change actions than for the slow speed and no lead vehicle scenario, and most probabilities are not simply 0, 0.5 and 1 as compared to the policy table for the conservative drivers. However, a significant proportion of the scenarios (8/18) still assigns equal probabilities for changes to the right and left lanes. In comparison, the aggressive policy recovered from the learned reward function has a substantially smaller proportion of scenarios (4/14) that are unable to distinguish between the changes to the right and left lanes. Overall, the neutral policy shows comparable behaviors for the expert-coded and learned reward functions. However, a few problematic actions are observed in the policy of the expert-coded reward function. For example, in the slow speed and no lead vehicle scenario, if the left distance is large and right distance is very small, the optimal actions of the expert-coded reward and the learned reward are to change to the right lane and left lane, respectively. To summarize, our method maximizes safety and yields more reasonable driving behaviors.

Finally, we test the prediction accuracy of the actions of all the four systems on the unobserved trips of other highway segments for the same set of drivers identified by our driving style identification method. We have 368, 398, and 229 shorter trips for aggressive, neutral, and conservative drivers, respectively. The results are shown in Tab. 2.7. Our proposed method outperforms the other three systems in the three categories of the testing

Table 2.6: Stochastic policy of the conservative drivers using the expert-coded reward function for slow driving without any lead car.

Dist. to Left Vehicles	Dist. to Right Vehicles	Probability of Action		
		Stay in Current Lane	Change to Right Lane	Change to Left Lane
No info	No info	0.592	0.408	0
No info	Very small	0	0.5	0.5
No info	Small	1	0	0
No info	Medium	0.05	0.95	0
No info	Large	1	0	0
Very small	No info	0	0.5	0.5
Very small	Very small	0	0.5	0.5
Very small	Small	0	0.5	0.5
Very small	Medium	0	0.5	0.5
Very small	Large	0	0.5	0.5
Small	No info	1	0	0
Small	Very small	0	1	0
Small	Small	1	0	0
Small	Medium	1	0	0
Small	Large	1	0	0
Medium	No info	0.069	0.078	0.853
Medium	Very small	0	1	0
Medium	Small	1	0	0
Medium	Medium	0.624	0.188	0.188
Medium	Large	1	0	0
Large	No info	1	0	0
Large	Very small	0	0.5	0.5
Large	Small	0.986	0.007	0.007
Large	Medium	0.996	0.002	0.002
Large	Large	0.931	0.035	0.035

Note: Blue shade denotes the optimal action

trips. More specifically, our customized automated lane change system with an aggressive driving style improved the prediction accuracy by 81.6%, 141.9% and 21.5%, respectively. Additionally, we visualize an example driving trip to highlight the difference in performance of our proposed method from the expert-coded reward function in Appx. A.2. In summary, our proposed method is capable of accurately predicting the driver actions for all the different driving styles, with best predictive power for the conservative drivers and largest

Table 2.7: Prediction accuracy for unobserved trips.

Methods	Prediction Accuracy (%)		
	aggressive	neutral	conservative
Non-customized	40.64	65.33	60.20
Customized using expert-coded rewards	30.52	61.67	74.4
Customized using GAIL	60.75	68.42	70.07
Customized using IRL (our method)	73.82	70.52	85.40

Note: The customized systems are tested for the corresponding trips with the same driving style.

improvement for the aggressive drivers.

2.6 Discussion

In this Chapter, we propose a two-step method for building customized automated lane change systems directly from naturalistic driving data. In the first step, our proposed method applies MFPCA for automated feature extraction and clustering analysis to identify driving styles. Our approach is able to reveal three distinct styles: aggressive, neutral, and conservative. The aggressive drivers tend to have the highest speed and smallest distance to the lead vehicles in open roads and the fastest acceleration when leaving the junction segment. On the other hand, the conservative drivers tend to decelerate very fast when entering the junction segment and maintain the lowest speed on the whole freeway segment and largest distance to the lead vehicles. Lastly, the neutral drivers have a smooth transition of speed when entering and leaving the junction segment. As for the open roads, the neutral drivers travel at speeds that are not as fast as the aggressive drivers but also not as slow as the conservative drivers. Our method, therefore, addresses the challenges of identifying driving styles quantitatively using driving behaviors. It also simplifies the interpretability of the results from unsupervised learning. In other words, minimal expert knowledge are needed to interpret the driving styles, which reduce excessive and potentially biased labeling efforts.

In the second step, we apply the IRL method to build customized lane change systems for each driving style. From the extracted optimal policies and reward functions, we observe that aggressive drivers prefer faster speeds, stay closer to the lead vehicles, and change lanes more frequently when compared to the neutral and conservative drivers. The neutral drivers prefer moderate speeds and do not stay too close or too far from the surrounding vehicles. They also adapt their speeds and locations by executing lane changes as required by their current situations. Lastly, the conservative drivers tend to minimize the number of lane changes by continuing in their current lanes and staying far away from the surrounding vehicles.

The performance of the customized lane change systems on unobserved trips was examined and compared to a non-customized system and two other customized systems. The results show the effectiveness of our proposed method and confirm that the learned reward function is more useful than its expert-coded counterpart. Our two-step method leveraged a clustering-based method to capture the behaviors of multiple drivers. Moreover, we improved the performance on real-world unobserved driving trips by using noisy naturalistic driving data instead of limited samples from a controlled driving environment. The value of using a two-step method is demonstrated through the computed rewards, which connects the identified driving styles in step one with the driving behaviors from the optimal stochastic policies in step two.

There are limitations of the proposed method. First, the naturalistic driving data can exhibit different driving behaviors for certain scenarios that are all within the safety threshold. For example, if an aggressive driver has to merge into one lane from a four-lane road, they need to take action regardless of their preferred driving style. This limitation can generate irrational behaviors in the training set, which will affect the reward values and optimal policies. However, the highest probabilities of the reasonable behavior (stay in the current lane) in the training set can, however, maximize safety when implementing the policy in future testing of the proposed systems. Second, the naturalistic driving data is purely observational and lacks experimental control. For this reason, some state transitions or state-action pairs may not be observed in the data. This exploration deficiency compromises the model performance.

Third, the computational cost of the IRL method can be huge for long trajectories.

Future studies can consider more realistic and complex scenarios that better account for the continuous state and action spaces. A more robust method may also have to be considered to overcome the irrational behaviors observed in the training set. Nevertheless, in general, our approach is promising for customizing automated lane change systems to driving styles. While the data used in this study does not provide examples of all the driving situations, we can expand on this approach to learn other meaningful driving behaviors.

Chapter 3

CUSTOMIZED HUMAN TRUST MODELS

Trust is an essential factor that affects the acceptance and reliance on automation. It has been widely studied in human-automation interaction. In this chapter, we introduced a customized real-time trust prediction model based on trust dynamics. The customized trust models provide implications for later human-automation integration design. This work has been published in *Proceedings of 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* [65].

3.1 Motivation and Objective

Vehicle automation technologies have significant benefits for society, including dramatic decreases in car crashes, injuries and deaths, increased mobility, increased road efficiency, and better utilization of parking and lands [42, 69]. Besides the advantages for society, they can also improve the driving experience and comfortableness of operations [33, 58]. Even though research has shown substantial benefits of vehicle automation technologies, their acceptance does not seem to keep up with the fast-growing market penetration. One of the most widely adopted methods to solve the acceptance issue is to calibrate trust in these technologies to the appropriate level since trust calibration is essential to accept and rely on vehicle automation [87]. Additionally, misuse of the system due to overtrust should be avoided [27]. Many studies have investigated the possible solutions to calibrate trust. These include paradigms that anticipate human behaviors—such as trust—and inform humans to make optimal choices [5, 32, 72]. However, a primary challenge for such an approach is quantitatively predicting human trust in real-time.

Most current studies use questionnaires to measure self-reports of drivers' trust levels

before, during, or after the interaction with the automated systems [24,51,96]. However, it is challenging to obtain self-reports repeatedly without interrupting the task. As an alternative, recent works have developed dynamic models to capture human trust and estimate it in real-time [5,18,80,122]. There are two approaches for developing such models: a general model for the whole population and a personalized model for each individual to account for individual differences [119]. A general trust model ignores individual differences but can be trained using limited data. On the other hand, a completely personalized model designed for each person requires a significant amount of data for each new user. Such a personalized model may be applicable for some small-scale systems. However, for broad commercial applications, the amount of training data required is more than that can be collected in a short time period. Therefore, a tradeoff exists between limiting the amount of data needed for model training and improvement in model performance by personalization.

The objective of this work is to address this tradeoff by using clustering methods to separate different trust dynamics across the sample population. We then develop customized trust models for each cluster of the population that account for broad individual differences in trust dynamics but allows model development with limited data.

3.2 Related Work

With the fast emerging of vehicle automation technologies, there is skepticism rising in public concerns [46]. In a 2013 survey, 66% of U.S. respondents indicated they were "scared" by the concept of automated driving, and more than half of respondents are skeptical of the reliability of the technology [104]. The results show significant disagreements, that some people are more confident about the future of automated systems and others are still skeptical, have arisen in the attitudes toward the automated systems. From the human factors perspective, the vehicle automation needs to identify drivers' psychological characteristics and cognitive processes because those factors are reported to influence how drivers use these technologies [29]. Trust has emerged as a relevant focus in research since it provides a solid foundation for describing the relationship between humans and automation [46, 90].

3.2.1 Trust understanding and modeling

Trust is shown to play a crucial role in understanding the acceptance of innovative technology [67]. In fact, [51] reported that trust determines the use or rejection of automation and willingness to rely on automation in certain situations. Researchers have pointed out important aspects of trust in human-machine interaction. Muir et al. [80] showed that an individual's mental model has a strong relationship with the way he/she trust in the system. The research also emphasized that trust changes through experiences in association with the change of his/her mental model of the system.

A well-accepted definition of trust in automation, proposed by Lee and See [61], is "the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability". They propose that the dynamic process rules trust and introduces context as a significant factor in trust development. Based on the definition of trust in [61], Hoff and Bashir [51] introduce three layers of trust: dispositional, situational, and learned trust. The dispositional trust is conceptualized based on early trust-related experiences that are typically affected by an individual's demographics. Situational trust is context-dependent and is affected by situational information. The learned trust evolves with the experience with the system and differs by individual's mental model. This work is widely referenced in the later studies since they consider the trust evolution as a dynamic process [23, 35, 99].

3.2.2 Trust estimation and individual differences

Most studies adopt trust-related questionnaires to obtain users' trust levels. Lee and Kolodge [60] applied a topic model-based clustering method to comments about consumer attitudes towards vehicle automation. However, the questionnaires have a delayed effect and ignore the trust dynamics. While the factors discovered in the study provide guidance on feature extraction of trust dynamics, we aim at capturing individual differences in trust dynamics to achieve real-time prediction of user trust for trust calibration.

Morra et al. [78] proposed using a combination of questionnaire and galvanic skin response signal to quantify trust. The experiment was carried out using virtual reality as a human-machine interface to convey situational information, which is shown to improve trust in vehicle automation. Akash et al. [3] proposed a customized set of psychophysiological features for each individual to build a classifier-based trust-sensor model, which has investigated the trust estimation in real-time. Nevertheless, the psychophysiological measurements are intrusive and impractical in real-world implementation. Several researchers have developed a variety of quantitative human trust models. These include regression models [28, 81], time-series models [52, 61, 76], and Markov models [34, 75]. Recent work has demonstrated the use of a partially observable Markov decision process (POMDP) to model human trust dynamics to improve human-robot performance [6]. Researchers have also used a state-space model to capture human trust dynamics while interacting with a Level 3 driving automation based on automation performance, drivers' gaze, and drivers' non-driving related task performance [18].

With the advancement of new methodologies, many new approaches have been proposed in recent years. Although there are studies that consider cultural differences that affect trust in automated systems [47], previous studies on quantitative trust modeling often disregard the "dispositional" aspect. In this work, we demonstrate that individual differences in trust behavior (i.e., dispositional trust) can be captured by clustering the participants based on their trust dynamics (i.e., how they gain/lose the trust) to improve the performance of quantitative trust models.

3.3 Online Study Design

To model and cluster the dynamics of human trust, we collected human subject data using an online study where the participants interact with a Level 2 driving automation. The study used a simulated autonomous driving recording that was prerecorded using a physical driving simulator. The study was deployed on Amazon Mechanical Turk [9], and the participants accessed the study online using their personal computers. During the study, the autonomous

car drove through a series of ten intersections in an urban environment. The participants could press the spacebar key on their keyboard to indicate their intent to take over if they did not feel safe with the driving. Along with their take-over behavior, participants were also asked to provide self-reports of their trust as well as the reliability of the automation after each intersection during the study. Additionally, they completed a 12-question 7-point Likert scale pre-study and post-study trust questionnaire adapted from [56].

Two within-subject factors were varied for the ten intersections: automation reliability and pedestrian presence. Automation reliability was defined only in terms of car's stopping behavior at an intersection for consistency. It had two levels: low reliability, where the car aggressively decelerates very close to the stopline (deceleration starting at < 25 meters), and high reliability, where the car smoothly decelerates toward the stopline (deceleration starting at > 60 meters). Note that the reliability of the driving automation can be varied by other factors as well. Pedestrian presence also had two levels: either pedestrians are present or absent at the intersection. The presence of pedestrians can increase the perceived risk by the drivers. The risk was randomly varied across the ten intersections to avoid any ordering effects.

Additionally, three factors that can affect human trust dynamics were varied between participants: scene visibility, overall reliability, and automation transparency. First, weather condition is shown to impact the trust levels significantly [101]. Scene visibility was varied by changing the weather of the environment and had two levels: high visibility where the scene was sunny and low visibility where the scene was foggy with snow. Second, situational characteristics such as automation reliability can also substantially impact trust dynamics [88]. In particular, users gain more trust in systems that are reliable. Overall reliability was designed to be affected by scene visibility such that the reliability of the automation reduces in low visibility scenes (as expected in real scenarios due to degraded scene perception). Specifically, the overall reliability had three levels: 100% reliable, where none of the intersections were of low reliability and only occurred during high visibility scenes; 80% reliable, where two intersections were of low reliability and occurred during both low and high visibility

Table 3.1: Eight drive types in the online study. Overall reliability is the percentage of high reliability operations.

Drv. Type	Overall Reliability	Visibility	Transparency
A	100%	High	High
B	80%	High	High
C	80%	Low	High
D	60%	Low	High
E	100%	High	Low
F	80%	High	Low
G	80%	Low	Low
H	60%	Low	Low

scenes; and 60% reliable, where four intersections were of low reliability and only occurred during low visibility scenes. The low reliability intersections were randomly chosen across the ten intersections. Third, studies have shown automation transparency also has a positive correlation with trust levels. With a higher level of transparency, the users have more access to the situational information leading to an increase in their trust [70]. Studies have shown changing the level of automation transparency can calibrate trust to the appropriate levels [6, 123]. In our study, during the high automation transparency, augmented reality (AR) cues are shown to the participant representing the driving automation's perception of the scene. The AR cues can provide vehicle speed information, navigation information, and object detection and prediction that are present in the scene. In low transparency, the object detection and prediction are not presented. Fig. 3.1 shows an example screenshot of the actual study scenario. Tab. 3.1 shows the resulting eight drive types and their corresponding characteristics. Tab. 3.2 shows the randomized distribution of reliability and risk in each drive type. Refer to the supplementary video for further demonstrations.

Participants: Two hundred thirty nine participants (121 males, 113 females, and 5 un-

Figure 3.1: An example screenshot of the study scenario of low visibility and high automation transparency. The bounding boxes highlight the signs, cars and other moving objects in sight.

known) with ages between 19 and 77 years (mean: 39 years) from the United States participated in and completed the study online. They were recruited using Amazon Mechanical Turk, with the criteria that they must live in the US and have completed more than 1000 tasks with at least a 95% approval rate. The compensation was $2.25 for their participation, and each participant electronically provided their consent. The Institutional Review Board at Purdue University approved the study. Each participant completed a randomly selected drive type from Tab. 3.1. Before the participants began the trial, they were given brief instructions about the study, and they completed a tutorial consisting of four intersections that helped familiarize them with the study interface. To ensure a uniform notion of trust across participants, they were explicitly informed about the definition adapted from [61] as follows:

"Trust is defined as your attitude that the self-driving car will help you achieve your goal of driving safely in a situation characterized by uncertainty and vulnerability."

Since the participants were not monitored during the study, we asked the participants

Table 3.2: Event configuration in each intersection. The crosses denote the low reliability operation, and the Ps denote the presence of pedestrians in the specific intersection.

Drv. Type	1	2	3	4	5	6	7	8	9	10
A		P			P		P	P	P	
B		P		P	X P	P		X	P	
C		P			X P	P	P	P	X	
D			X	X P	X P		P		P	X P
E					P		P	P	P	P
F	P		P	P			P	P	X	X
G	P					X P	P	P		X P
H	P	P		X	P	X	P	X	X P	

to complete the study in fullscreen mode to avoid distractions. To avoid non-complying participants, we tracked the key-presses on the keyboard during the trial and removed the participants from the dataset who were suspected of exiting the fullscreen mode during the study. Furthermore, we removed the participants from the dataset who had missing survey data. As a result, 40 participants' data were removed from the dataset.

To summarize the online study design, we collected real-time trust levels of the users along with their take-over behavior while interacting with a Level 2 driving automation in a simulated environment. The study design considers three drive-level factors that can significantly impact the drivers' trust dynamics. Moreover, two event-level factors are considered within each drive type. The collected data will be used to identify the key characteristics of users trust dynamic, which will be the basis of clustering users based on their trust behavior.

3.4 Methods

Using the intersection-by-intersection trust measurements obtained from the participants, we analyze the trust dynamics of each individual. We observe significant variations across individual trust behavior. Specifically, some participants start with lower initial trust levels as they may be skeptical about the automation. As they interact with a consistently high reliability automation, their trust levels increase gradually. However, if they encounter a low

reliability operation, their trust levels drop drastically. As an extreme case, some participants' trust levels remain low throughout the study and do not increase significantly even after experiencing high reliability operation. On the contrary, some participants start with high initial trust levels and maintain stable and high trust levels throughout the study.

3.4.1 Trust Evolution Decomposition

We visualized the collected trust data to gain insights into the trust dynamics. Since there are drive types that do not have the low reliability operations or the low reliability operations happen at the end of the drive, we remove the participants from those drives to ensure all the participants for analysis have encountered the similar events in the study. As a result, we removed drive type A, E, and F from the samples, which left with 138 participants for further analysis. Based on the observations from the participants' data, we noticed the trust dynamics had three general phases across all the participants. The first phase is the initial *trust-building* phase. In this phase, the participants start from their initial trust levels and gradually gain some trust as they experience high reliability operations. The initial trust levels depend on the personal characteristics of the participants. For example, people who have indicated prior good experience with advanced vehicle automation systems in the pre-study survey (scores are greater than 5 out of 7) are likely to have a higher initial trust level (average initial trust level is 88.6). The second phase is the *error-awareness* phase, which occurs when a participant encounters the low reliability operation of the automation. After observing that the automation is not perfect, participants typically lose their trust in automation drastically. Finally, the third phase is the *trust-repair* phase. This phase follows the error-awareness phase during the interaction, where the participants regain their trust in the automation as they experience consistent high reliability operations after low reliability ones. The trust increase during the trust-repair phase is typically lower than the initial trust-building phase since the participants realize that the automation may not be perfect and is prone to errors.

For example, Fig. 3.2 shows the average trust dynamics for drive G. In this drive, there

are two low reliability operations at intersections 6 and 10. We observe that participants typically start with a relatively high average initial trust level (~ 77). This is consistent with findings that recent widespread use of automation has led to humans trusting a system when they have no experience with it [71]. During the initial trust-building phase (i.e., during consistent high reliability operation till intersection 5), the average trust level gradually increases for most of the participants, as apparent from smaller confidence intervals. At intersection 5, the trust level has the smallest confidence interval with the highest average value for this drive. Then in the two error-awareness phases (intersection 6 and 10), we observe a significant decrease in the trust levels and wider confidence intervals compared to the previous intersection, respectively. In this phase, participants notice the low reliability operations, which significantly reduces their trust level. Moreover, the first low reliability operation results in a much more significant effect in the decrease of trust level. Finally, the trust-repair phase (intersection 7 to intersection 9) shows the participants gradually regaining their trust after the low reliability operation if no further low reliability operation occurs. However, the rate at which the trust increases during the trust-repair phase is not as large as during the initial trust-building phase. Similar trends are also observed in other drive types.

In summary, we observed three main phases of the trust dynamics during the interaction with imperfect automation. However, the wide confidence intervals in Fig. 3.2 also suggest a considerable variation of the participants' trust levels across the three phases. In particular, the wide confidence intervals for the initial trust level show the participants have significant initial differences toward the automation. Although the confidence intervals converge during the trust-building phase, the confidence intervals are still wide during the error-awareness and trust-repair phases. These observations suggest that a general model may not be effective in capturing the individual differences in trust dynamics, and there is a need to develop customized models of trust.

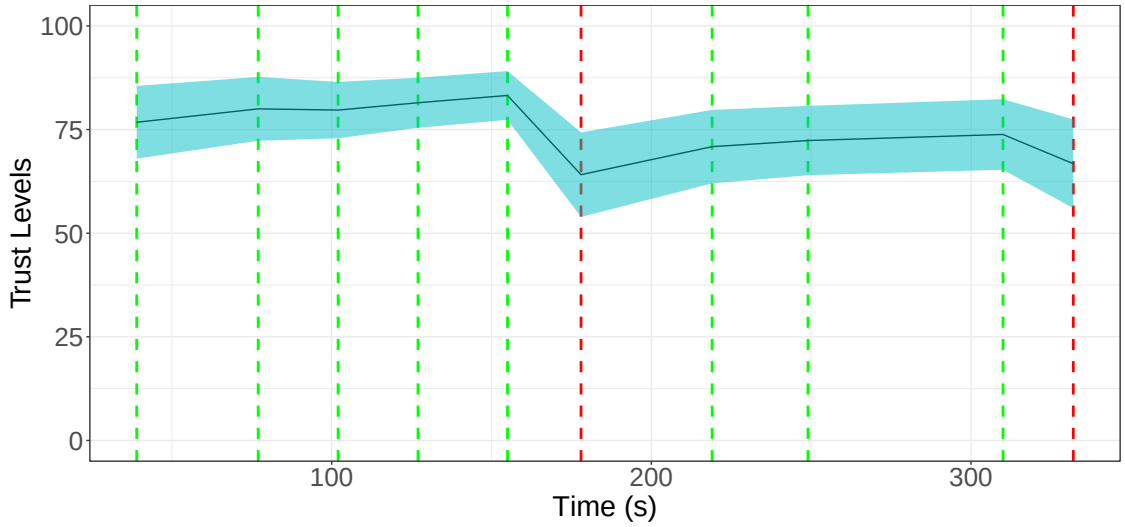


Figure 3.2: Average trust dynamics for drive G. Each dashed vertical line denotes an intersection in the study. The green lines represent the high reliability operation, and the red lines represent the low reliability operation in that particular intersection. The black solid line denotes the average trust levels across all participants for drive G with the blue shaded area denoting the 95% confidence intervals.

3.4.2 Trust Behavior Clustering

To cluster the participants based on their trust dynamics, we first extract features characterizing the dynamics and behaviors of participants' trust throughout each interaction for each individual. Specifically, we extract the rate of change of trust to capture the trust dynamics for each of the three trust evolution phases. Furthermore, based on our preliminary analysis of trust, we observe that trust is strongly correlated to the presence of pedestrians at the intersection as well as participants' take-over response. Therefore, for the trust-building and trust-repair phase, we extract the average trust of participants at intersections where: 1) pedestrians are present, 2) pedestrians are absent, 3) the participants take-over 4) the participants do not take over. We additionally consider the initial trust level as a feature as it can capture the individual differences we observed in our analysis. This results in 12

Table 3.3: List of 12 extracted trust dynamics features.

Phase	Feature
Trust-building	Initial trust level
	Rate of change of trust
	Average trust with pedestrian present
	Average trust with pedestrian absent
	Average trust with take-over
	Average trust with no take-over
Error-awareness	Rate of change of trust
Trust-repair	Rate of change of trust
	Average trust with pedestrian present
	Average trust with pedestrian absent
	Average trust with take-over
	Average trust with no take-over

features in total as listed in Tab. 3.3.

Considering the limited sample size of unique participants[1], we use simple Euclidean distance-based clustering method. To minimize the 'curse of dimensionality' [17] for the clustering algorithm, we use principal component analysis (PCA) to reduce the number of features used for clustering [120]. We apply PCA on the extracted features in Tab. 3.3 to reduce the dimension of data and provide insights on significant features that contribute most to the total variation. The explained variance ratios for the first three PCs are 44%, 17%, and 11%, respectively [2]. Thus, the first 3 PCs explain about 72% of the total variance

[1]Although our data is not small to model trust behavior as each participant contribute to multiple trust samples, the number of unique participants (138) is limited for data-based clustering.

[2]The variance ratios for the 4th and the subsequent components are 9%, 6%, etc.

in the extracted features.

Finally, we use the K-means clustering method to find the groups of people with similar trust behaviors. We chose the number of clusters as two since the silhouette analysis shows two clusters have the largest average silhouette coefficient (0.45) among all numbers of clusters varying from two to six and it ensures the best interpretability with significant statistical difference in all extracted features. With the identified clusters of participants, we will train customized models for each cluster to capture the individual differences of trust dynamics in each group. Furthermore, a close look into the clusters can provide insights into group-specific similarities.

3.5 Results and Validation

We applied the K-means clustering algorithm on the first three PCs to generate two clusters. The resulting two clusters have 36 and 102 participants, respectively. Fig. 3.3 shows the boxplots that demonstrate the variations in four representative features for the identified two clusters. For the initial trust (Fig. 3.3(a)) and the average trust with pedestrian absence during trust-building (Fig. 3.3(b)), the two clusters show a statistically significant difference based on two sample t-test with a significant level of 0.05. Specifically, the cluster shown in orange has relatively lower initial trust than that shown in blue. Furthermore, the orange cluster also has a lower average trust with pedestrian absence during trust-building than the blue cluster. This shows that the participants comprising the orange cluster are more skeptical than those comprising the blue cluster. They tend to have a low initial trust toward the automation as well as a low trust level even during a low risk (due to the absence of pedestrians) and high reliability operations. Therefore, we name the orange cluster as the "skeptical" group due to their lack of trust. On the contrary, we call the blue cluster the "confident" group as they show high confidence on the driving automation. In the error-awareness phase, we observe from Fig. 3.3(c) that the "skeptical" group's trust levels are more volatile compared to the "confident" group and thereby drop faster after they encounter a low reliability operation. Finally, during the trust-repair phase, we consider the average

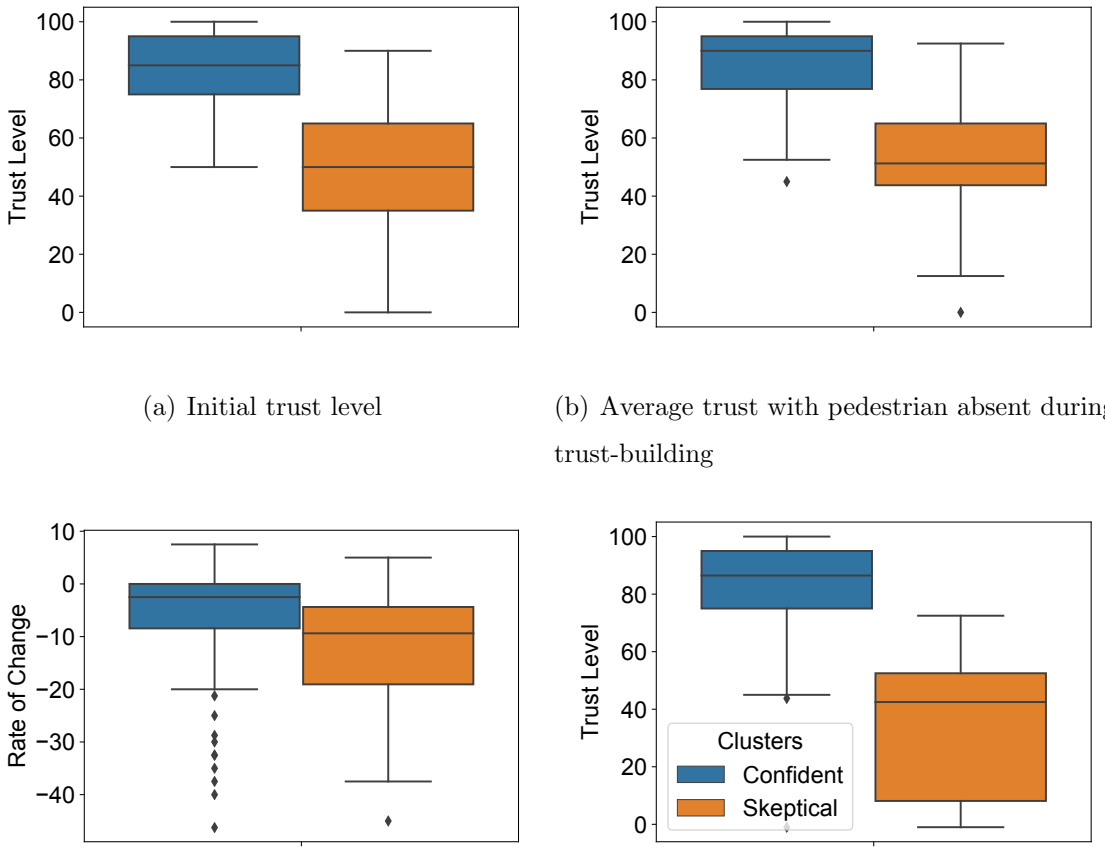
trust level with no take-over (Fig. 3.3(d)). This feature represents the trust level when the participants are comfortable with the automation system. We observe that the "skeptical" group has statistically lower trust levels than the "confident" group even when they do not take over. A two group t-test shows all 12 features are significantly different for the two clusters.

To better demonstrate the characteristics of the trust dynamics for the two groups, we compare the evolution of average trust for same drive type G as Fig. 3.2in Fig. 3.4. The data for this drive type comprise of 22 "confident" group participants and 5 "skeptical" group participants. We see that the "skeptical" group start with a relatively lower initial trust and their trust level drops more quickly during a low reliability operation compared to the "confident" group.

While these clusters were identified using data-driven techniques, it is important to verify these behaviors with cognitive and behavioral psychology literature. Prior studies of both interpersonal human trust as well as human trust in automation note the existence of distinct trust behaviors among humans. Studies using the Rotter Interpersonal Trust Scale [93] have found two groups of trust behaviors, namely, "high trusters" and "low trusters". Furthermore, it was noted that characteristics of each individual varied in terms of willingness to trust a novel situation. Though the differences between humans' trust in automation was not explicitly analyzed, the other metrics established the two groups, such as high trusters being more willing to trust experimenters [94, 95]. It is likely that the "confident" and "skeptical" partially represent the high and low trusters, respectively.

3.5.1 Customized Real-time Trust Prediction

We show the improvement in trust prediction performance using customized models as compared to general model. The general model is trained using data for all participants and therefore, ignores the individual trust variations in trust dynamics. The customized models are separately trained using data from each group of participants identified by clustering, respectively. The resulting customized models leverage the distinct trust behaviors of each



(a) Initial trust level

(b) Average trust with pedestrian absent during trust-building

(c) Rate of change of trust during error-awareness

(d) Average trust with no take-over during trust-repair

Figure 3.3: Boxplots of four representative features of trust dynamics. The two clusters show significantly difference from each other in all four features.

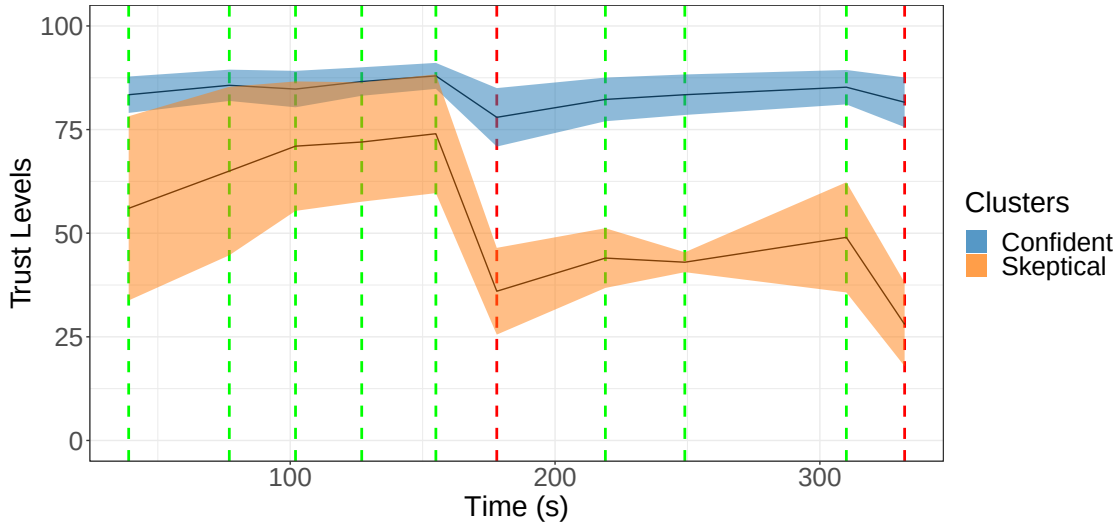


Figure 3.4: Average trust level of the "skeptical" and the "confident" group participants at each intersection for drive G. The shaded area denotes the 95% confidence intervals of the trust level.

group that significantly improve the models performance. We consider two structures of trust prediction models for this comparison.

Linear Regression (LR) Model:

We first consider a simple linear regression model for predicting users' trust. We use the observations from the previous two intersections to predict the trust level at the current intersection. Specifically, we consider scene visibility, automation transparency, presence of pedestrians, automation reliability, participants trust level, and participants take-over behavior in the previous two intersections as an input to the model to predict the current trust level. Although this model may not be practical if self-reports of trust are unavailable, the models provides a simple baseline to validate the clustering performance.

State Space (SS) Model with Kalman Filter:

A classical approach to model a dynamical system is by using a linear time-invariant state-space (SS) model. Linear SS model has been used to capture human trust dynamics while interacting with a Level 3 driving automation based on automation performance, drivers' gaze, and drivers' non-driving related task performance [18]. Considering trust as a continuous state of the system, they use Kalman filter to estimate trust during the interaction. Since our output of take-over intent is a binary variable, we adapt the linear state space model in [18] with a sigmoid output function that maps the state of trust to the output of take-over. Thereby, we consider scene visibility, automation transparency, pedestrian presence, and automation reliability as inputs of the SS model; the state is a continuous variable of trust; and the output is the take-over intent. The model parameters are estimated using linear mixed-effect model with participants as a random effect as described in [18]. Note that the self-reported trust is used as the measurement for the continuous trust state and is only used for model training. To evaluate the prediction performance of trust and take-over intent, we use an extended Kalman filter (EKF) that accommodates the nonlinear sigmoid output function to update the state estimate of trust after each output is observed, which is then used to predict the next take-over intent based on current inputs. Therefore, the EKF provides a real-time estimate of trust and take-over without the need for self-reports of trust during real-time prediction. The model is characterized by (3.1).

$$\begin{aligned} T_{k+1} &= \mathbf{A}T_k + \mathbf{B}\begin{bmatrix} v_k & t_k & p_k & f_k \end{bmatrix}^T \\ b_k &= \mathrm{Sig}(\mathbf{C}T_k + \mathbf{C}_b) \end{aligned} \tag{3.1}$$

Here $\mathrm{Sig}(x) = (1 + e^{-x})^{-1}$, T_k is the trust level at the kth event, v is the scene visibility, t is the automation transparency, p is the presence of pedestrians, f is the automation reliability, b is the take-over behavior of the participant, and $\mathbf{A}, \mathbf{B}, \mathbf{C}, \text{and}, \mathbf{C}_b$ are linear parameters.

Additionally, we compare the performance of our proposed trust dynamics-based clustering with demographic information based clustering. Specifically, we consider three baselines: 1) age with threshold as the mean age (40 years) of the participants sample; 2) gender (male

Table 3.4: MSE and F1 scores for general model and customized models using different clustering criteria. Lower MSE and higher F1 score indicates better model performance.

Clustering Criteria	Cluster	Number of participants	MSE for trust		F1 scores for take-over
			LR model	SS model	SS model
-	General model	138	0.602	0.518	0.423
Trust dynamics (our method)	"Confident"	102	0.442	0.381	0.468
	"Skeptical"	36	0.384	0.289	0.650
Age	"At least 40"	53	0.587	0.586	0.520
	"Less than 40"	85	0.592	0.431	0.327
Gender	"Male"	65	0.594	0.488	0.456
	"Female"	71	0.526	0.549	0.396
Driving Style	"Aggressive"	61	0.543	0.571	0.347
	"Conservative"	77	0.526	0.493	0.460

or female); and 3) self-reported driving style (aggressive or conservative). We perform a 5-fold cross validation (CV) with uniform distribution of each drive type in the training and validation sets for both the model structures to obtain the validation performances. We calculate the validation mean squared errors (MSE) for trust prediction using both models and the F1 scores for take-over intent prediction using the SS model for the general model as well as for the clusters using each of the clustering criteria. The result is shown in Tab. 3.4.

We see that for both the LR and SS models, the customized models based on the trust dynamics-based clustering performs better than the general model. That is, the trust dynamics-based customized models have lower MSE for trust as well as higher F1 score for take-over intent prediction. Therefore, the customized models successfully improves the trust prediction by considering the individual differences across the population. However, we do not see such significant improvement for age-based, gender-based, or driving style-based customized models. To further quantify the improvement in the prediction performance us-

Table 3.5: Percentage increase in the prediction performance using the customized model as compared to the general model

Clustering Criteria	MSE for trust		F1 score for take-over
	LR model	SS model	SS model
Trust dynamics	29.1%	31.1%	21.7%
Age	2.0%	5.2%	-5.2%
Gender	7.1%	-0.3%	0.5%
Driving style	11.3%	-1.7%	-3.1%

ing the customized model, we calculate the percentage increase in the average metrics (MSE and F1 score) for each clustering criteria as compared to the general model. The average metrics for each clustering criteria is calculated as the mean of the metrics across the clusters weighted by the number of participants in each cluster. The resulting improvements in the prediction performance is shown in Tab. 3.5. We observe that the trust-dynamics based customized models not only improves the prediction performance, it significantly outperforms simple demographic factor based clustering. This shows that these demographic factors alone may not be strong contributors to the variations in human trust behavior.

In summary, we demonstrate that trust dynamics-based clustering can allow to develop improved trust model needed for trust calibration paradigms. In practice, a small interaction data from a user can be used to determine the cluster to which the user belongs to and accordingly utilize the pre-trained customized models and policies for the given cluster. This allows ease of deployment in commercial settings without the need to retrain the models for personalization.

3.6 Discussion

We presented a trust dynamics-based clustering framework to identify and develop customized trust models based on dominant human trust behavior among a large population. We showed that such a framework can balance the tradeoff between a single general, or several personalized, models of human trust. We identified participants in the two clusters, namely "skeptical" and "confident" based on their trust behavior. We showed that customized models developed based on these clusters significantly outperforms a general *one-fit-all* model in predicting human trust and take-over behavior during interaction with a driving automation. Furthermore, trust dynamics-based clustering approach is better than age-, gender-, or driving style-based approach in developing such customized model. Finally, we showed that the clustered participants' behaviors could be explained reasonably and may be coincident with established psychology of human trust. Future work could involve using these customized models for real-time trust calibration paradigms to improve human-automation interactions.

Chapter 4

TRUST-CALIBRATED CUSTOMIZED VEHICLE AUTOMATION

In this chapter, we integrate the real-time trust prediction model and the customized vehicle automation using the human-in-the-loop method to achieve trust calibration. Our proposed IRL-SS algorithm provides an algorithm to incorporate human cognitive feedback into the training process of the automated agent.

4.1 Motivation and Objectives

Prospective vehicle automation is expected to provide customization and dynamic response to changes in drivers' cognitive states for better acceptance and reliance. The human-in-the-loop system design method is essential for building such systems. In the previous chapters, we introduced customization in vehicle automation and customized real-time trust prediction model. We are one step away from our general research goal of this dissertation: to effectively integrate the trust models into the customized vehicle automation to achieve trust-calibrated customized automated systems.

Trust as a significant factor that affects the reliance on vehicle automation has been widely studied in previous studies. In recent years, due to the agile advancement of automated systems, trust calibration has drawn a lot of attention [109, 126]. However, most studies calibrate trust by adjusting the level of transparency using POMDP [5, 6]. This limits the design paradigm to only changing the interface of the automated systems. Another gap in current literature is that researchers rely on psychophysiological measurements for real-time trust sensing and estimation [3]. These methods can interrupt the continuity of interaction between humans and automation, and the intruding nature of psychophysiological

measurements is impractical in real-world testing and implementation.

The objective of this work is to integrate the human trust models and customized automated systems to achieve non-interrupting and non-intruding real-time trust prediction for human-aware system design.

4.2 Related Work

Human-system integration is broadly related to incorporating human feedback into the training process of automated agents using the human-in-the-loop method. Interactive Learning methods are widely studied in Human-computer Interaction (HCI) and Human-robot Interaction (HRI) communities. Specifically, if the automated agents have the decision-making capability, Interactive RL is often adopted.

In Interactive RL, the agent learns from human evaluative feedback, such as evaluations of the quality of the behavior, advice, or instruction [64]. The Interactive RL can be categorized into different types based on design dimensions and type of feedback from human users [16]. Reward shaping is the most popular design dimension that guides agent training by adding additional rewards from human feedback directly to the original RL reward functions [14,85]. Some applications also use human feedback as the only reward function in the training process [15, 57]. Although the reward shaping method is straightforward in agent training, it heavily depends on a sophisticated reward design. In many real-world applications, the complexity of the decision process makes the reward design very difficult.

Another design dimension is the policy shaping method [41]. The policy shaping method consists of formulating human feedback as action advice that directly update the agent's behavior [16]. Compared to reward shaping, policy shaping does not rely on the design of the reward function. Therefore, the agent can directly extract the (near-)optimal action rather than inferring it using the underlying reward function. However, the policy shaping methods require a substantial amount of human feedback to generate decent policies.

The next design dimension is the guided exploration process which can mitigate the sample-inefficient issue of the basic RL algorithms. It aims to minimize the learning proce-

dure by injecting human knowledge that guides the agent's exploration process [125]. The guided exploration process biases for actions that have a high probability of being near-optimal but avoids exploring the actions that potentially lead to non-optimal policies. One popular approach to guiding the exploration process consists of creating myopic agents [36]. This kind of shortsighted agent constructs its policy by choosing at every time step to perform the action that maximizes the immediate reward, according to the action suggested by the user; the action's long-term effects are not considered. However, this approach trains agent that tends to overfit [16].

The last design dimension is to augment the value function. It is similar to reward shaping but combines the value functions instead of reward functions from humans and agents. This approach was shown to accelerate the learning rate [21]. Nonetheless, the reuse of value functions might be a good way to minimize the human feedback required.

The above design dimensions provide a general approach to integrating human feedback into the RL framework. Another research direction that differs from the design dimensions is studying how to deal with different types of feedback. The users can provide binary critique [54], scalar-valued critique [62], action advice [10], etc. However, these frameworks require a considerable amount of active interactions initiated by humans. Since trust is a hidden cognitive state, it is challenging to inquire and sense. Moreover, actively inquiring about human trust can result in fatigue and biased response. Therefore, passively estimating and predicting trust is ideal for human-aware automation. In summary, the goal is to find a sample efficient integration method to connect the trust models with the RL framework.

4.3 Online Study Design

To collect trust dynamics while interacting with our customized vehicle automation, we designed an online driving simulator study using Carla Simulator [31]. The driving environment is a four-lane highway driving scenario with a 60 mph speed limit. The weather and time are clear noon. The study was also deployed on Amazon Mechanical Turk [9], and the participants accessed the study online using their personal computers. During the

Figure 4.1: An example screenshot of the study scenario in Carla Simulator.

study, the autonomous cars sample 15 control actions from the three policy tables of different driving styles generated in customized vehicle automation step. The participants could press the arrow keys to indicate their intent to take over if they did not feel safe with the lane change behaviors. Along with their take-over behavior, participants were also asked to provide self-reported trust levels after each lane change action. Additionally, they completed a 12-question 7-point Likert scale pre-study and post-study trust questionnaire adapted from [56].

The 15 lane change behaviors consists of five aggressive lane changes, five neutral lane changes and five conservative lane changes sampled from the learned policy tables using MaxEnt IRL, respectively. Additionally, we also collect basic demographic information including driving styles (very aggressive, aggressive, neutral, conservative, very conservative) for further analysis of the customized systems. Fig. 4.1 shows an example screenshot of the actual study scenario.

Participants: Since the participants were not monitored during the study, we asked the participants to complete the study in fullscreen mode to avoid distractions. To avoid non-

complying participants, we tracked the key-presses on the keyboard during the trial and removed the participants from the dataset who were suspected of exiting the fullscreen mode during the study. After data cleaning to exclude the participants who did not follow the study protocol, 99 participants (63 males, 36 females) with ages between 25 and 60 years (mean: 39 years) from the United States participated in and completed the study online. They were recruited using Amazon Mechanical Turk, with the criteria that they must live in the US and have completed more than 1000 tasks with at least a 95% approval rate. The compensation was $3 for their participation, and each participant electronically provided their consent. The Institutional Review Board at the University of Washington approved the study. Each participant completed the study with 15 lane change actions. Before the participants began the trial, they were given brief instructions about the study, and they completed a tutorial consisting of four lane change behaviors that helped familiarize them with the study interface. To ensure a uniform notion of trust across participants, they were explicitly informed about the definition adapted from [61] the same as the previous online study. The trust dynamics of an example participants is show in Fig. 4.2.

To summarize the online study design, we collected real-time trust levels of the users along with their take-over behavior while interacting with our previously designed customized automated lane change systems in a simulated environment. The study design considers three driving styles that behave differently in the same environment settings.

4.4 Methods

Study has shown self-reported driving styles have modest but significant correlations to the actual driving styles [30]. Therefore, to match the drivers with appropriate customized vehicle automation using the self-reported driving styles can ensure that they are riding in an automated systems that behave exactly like them. However, the imitation of the behavior is not perfect. Specifically, the prediction accuracy of the actions is under 75% for aggressive and neutral drivers. Moreover, they are blind to human trust changes and maintain their constant behaviors all the time. Therefore, we proposed an IRL-SS algorithm to integrate our

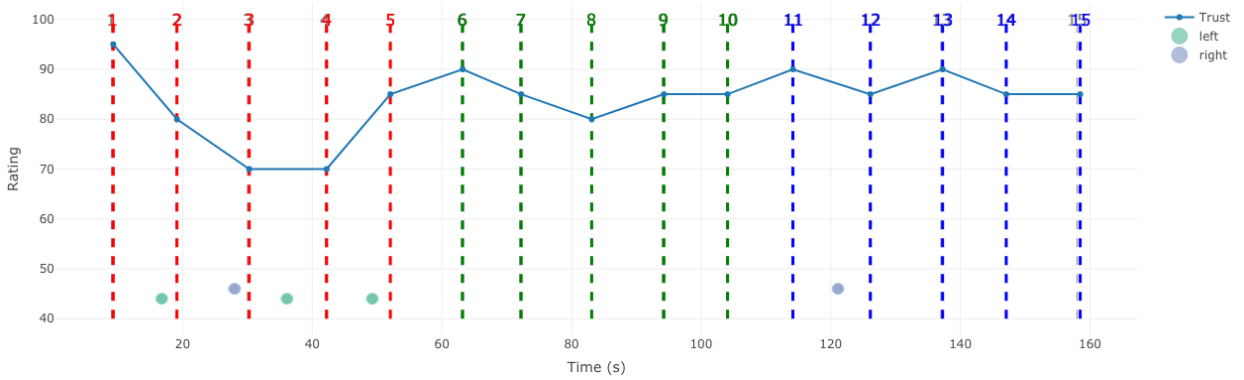


Figure 4.2: Trust dynamics of an example participant. The trust values of the pre-study and post-study survey are 4.6 and 5.8, respectively. The self-reported driving style is neutral. The dashed lines indicate the 15 lane change actions in the time horizon (x axis). The blue piecewise linear line denotes the trust scores (y axis). The red, green and blue colors denote the aggressive, conservative and neutral lane change actions, respectively.

customized real-time trust models into the IRL framework to achieve better performance. The overall framework is shown in Fig. 4.3. Here, trust feedback is estimated by trust prediction models passively, and serves as additional reward in the training process of the agent.

4.4.1 Customized Trust Models

Our online study shows that trust dynamics is a significant factor that affect the model performance. Therefore, before building the trust models, we first cluster the participants to three "driving-trust" styles based on their trust dynamics. In the previous chapter, we developed a linear SS model to predict real-time trust and take over intent. The general form of the trust model can be characterized as follows:

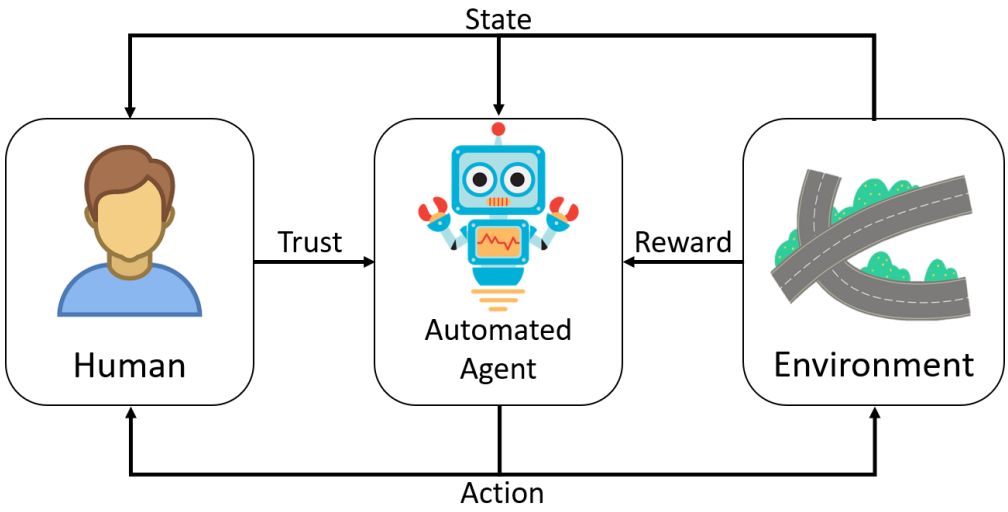


Figure 4.3: Interactive RL framework. Human is the new entity in the original RL framework. Human sense environment states, receive actions generated from agent, and provide their trust as additional feedback to train the agent.

$$T_t = h(T_{t-1}) + g(u_{t-1}) + \epsilon_{t-1} \tag{4.1}$$

$$y_{t-1} = f(T_{t-1}) + \nu_{t-1} \tag{4.2}$$

where T_t is the trust levels at time t, a_{t-1} is the input at previous time step, y_t is the output, $h(\cdot), g(\cdot), f(\cdot)$ are fitted functions, and ϵ, ν are random noises.

We still model our output of take-over intent as a binary variable, we adapt the linear state space model in [18] with a sigmoid output function that maps the state of trust to the output of take-over. Thereby, we consider environment state, lane change actions as inputs of the SS model; the state is a continuous variable of trust; and the output is the take-over intent. The model training and evaluation is the same as previous chapter. The model is characterized by Eq. (4.3).

$$
\begin{aligned}
T_{t+1} &= \mathbf{A}T_t + \mathbf{B}\begin{bmatrix}\mathbf{s}_t & a_t\end{bmatrix}^T \\
b_t &= \mathrm{Sig}(\mathbf{C}T_t + \mathbf{C}_b)
\end{aligned}
\tag{4.3}
$$

Here $\mathrm{Sig}(x) = (1 + e^{-x})^{-1}$, T_t is the trust level at time-step t, $\sim_t$ is the environment state from the MDP, a_t is the optimal action, b is the take-over behavior of the participant, and $\mathbf{A}, \mathbf{B}, \mathbf{C}, \text{and}, \mathbf{C}_b$ are linear parameters.

4.4.2 Proposed IRL-SS for Reward Shaping

We propose a new reward shaping method by integrating real-time trust prediction model with the IRL algorithm. Our method differs from the previous methods in term of the types of feedback that we obtained from humans. Previous studies mainly consider active feedback such as guidance, action advice, binary critiques, etc. which require the users to actively provide their response during the interaction. These types of feedback can cause serious fatigue or unwillingness of sharing due to the long course of interaction such as driving tasks. Moreover, self-reported trust can be biased if the users are asked about the same trust questions constantly. To mitigate the above issues, we proposed to adapt reward shaping in our customized systems to build a trust-calibrated system which can better capture their driving behavior and promote trust. The reward shaping is given as follows:

$$
R_t = w_1 r'(T_t) + w_2 r_t(s_t, a_t) \tag{4.4}
$$

where $w_1 + w_2 = 1$. Here, R_t is the total reward in time-step t, w_1 and w_2 are the weights, $r'(T_t)$ is the reward mapping from trust level T_t to a scalar reward value, and $r_t(s_t, a_t)$ is the learn reward from IRL algorithm.

With Eq. 4.4, we can derive the proposed IRL-SS algorithm using the new combined reward. Now the reward of a trajectory becomes:

$$
R'(\zeta) = w_1 \sum_{\zeta} r'(T_t) + w_2 \sum_{s \in \zeta} \theta^T f_s \tag{4.5}
$$

And based on the principle of maximum entropy, the probability of a trajectory is still proportional to the exponential of the reward of the trajectory given in Eq.2.10.

Therefore, the gradient of the likelihood function is,

$$\begin{aligned}\nabla_\theta L(\theta) &= \frac{w_2}{M}\sum_{\zeta_D} f_{\zeta_D} - \frac{1}{\sum_\zeta e^{R'(\zeta)}}\sum_\zeta (e^{R'(\zeta)}\frac{dR'(\zeta)}{d\theta}) \\ &= \frac{w_2}{M}\sum_{\zeta_D} f_{\zeta_D} - w_2\sum_\zeta \frac{e^{R'(\zeta)}}{\sum_\zeta e^{R'(\zeta)}} f_\zeta \\ &= \frac{w_2}{M}\sum_{\zeta_D} f_{\zeta_D} - w_2\sum_\zeta P(\zeta|\theta) f_\zeta \\ &= \frac{w_2}{M}\sum_{\zeta_D} f_{\zeta_D} - w_2\sum_s P(s|\theta) f_s\end{aligned} \tag{4.6}$$

where $P(s|\theta)$ is the state visitation frequency given the reward parameter θ. Finally, Alg. 3 Alg. 2 Alg. 4 can be revised corresponding to the above gradient.

Algorithm 5: Revised Value Iteration

for $k = 1, 2, \ldots, \infty$ **do**
  for *each state* s **do**
    $V_k(s) = \max_a \sum_{s'} P_a(s, s')(R'(s) + \gamma V_{k-1}(s'))$
  if $\forall s, |V_k(s) - V_{k-1}(s)| < threshold$ **then**
    for *each state* s **do**
      $\pi(s) = \arg\max_a \sum_{s'} P_a(s, s')(R'(s) + \gamma V_{k-1}(s'))$ **return** π, V_k
  else
    continue

Algorithm 6: State Visitation Frequency Calculation

Solve the MDP using Alg. 5 for optimal policy π;

Computer $\mu_1(s)$ using sampled trajectories;

for $t = 1, 2, \ldots, T$ **do**
  $\mu_{t+1}(s|\theta) = \sum_a \sum_{s'} \mu_t(s')\pi(a|s')P_a(s, s')$
return $P(s|\theta) = \sum_t \mu_t(s|\theta)$

Algorithm 7: Revised IRL algorithm

0. Initialize θ randomly;
1. Solve for optimal policy π w.r.t θ with Alg. 5;
2. Solve for state visitation frequency $P(s|\theta)$ using Algorithm 6;
3. Compute gradient $\nabla_\theta L(\theta) = \frac{w_2}{M} \sum_{\zeta_D} f_{\zeta_D} - w_2 \sum_s P(s|\theta) f_s$;
4. Update θ with one gradient step using $\nabla_\theta L(\theta)$

Since the linear SS model for real-time trust prediction does not have the feedback loop to depend on the IRL algorithm, it does not have to change and can be trained independently. This is based on the assumption that the trust model is not a moving target, so we can model and predict real-time trust using environmental factors and human observable behaviors.

4.5 Results

4.5.1 Clustering and Trust Modeling

We cluster the participants based on their trust dynamics and self-reported driving styles. Nine features are used in the clustering step including:

1. average trust levels for aggressive, neutral, and conservative lane change style,
2. rate of change of trust for aggressive, neutral, and conservative lane change style,
3. trust scores of pre-study surveys,
4. trust scores of post-study surveys, and
5. self-reported driving styles (1=very conservative to 5=very aggressive).

Three clusters were selected and compared to the three customized trust models with the IRL algorithm. Our goal was to match the trust dynamics over time with driving style. As a result, 42 participants were identified as "aggressive" group with an average driving style score of 2.88 (90% CI:[2.70, 3.07]), 42 participants were identified as "neutral" group with an

average driving style score of 2.24 (90% CI:[2.08, 2.40]), and 15 participants were identified as "conservative" group with an average driving style score of 2.88 (90% CI:[1.65, 2.21]).

We also observe that the "conservative" drivers are more skeptical toward the automated lane change systems. Although the initial trust levels are comparable to the other groups, the average trust trend is below the other two groups and has a wide CI. Especially, during the first five aggressive lane changes, the "conservative" drivers tend to loos trust very fast and maintain a relatively low trust levels for the entire study. On the contrary, the "neutral" drivers are very confident about the automated systems since they have a very stable and high trust levels in the entire study, and the CI is very narrow. Although they also have a small drop in trust levels in the first three lane changes, they recovered to a even higher trust level in the rest of the study. Finally, the "aggressive" drivers maintains a relative stable and high trust levels across the whole study. They have slightly lower trust levels than the "neutral" drivers, but the trend is increasing steadily at around 80 trust levels. These observations show that the "aggressive" drivers are neither overtrust nor undertrust toward the automated lane change systems but maintain a vigilant attitude toward the automated systems.

Henceforth, based on the trust dynamics of each cluster, we further classify each group as "Aggressive-Vigilant", "Neutral-Confident", and "Conservative-Skeptical", respectively. The final names and clusters are visualized in Fig. 4.4.

After we successfully cluster the participants into different groups, we can build the corresponding customized trust models. We build three SS models for each cluster of participants. Similarly to the previous analysis step, we compare the customized models with general model and customized based on self-reported driving style models. The result in Tab.4.1 demonstrates the improved performance of our proposed customized models which shows the effectiveness of clustering participants based on trust-dynamics and self-reported driving styles.

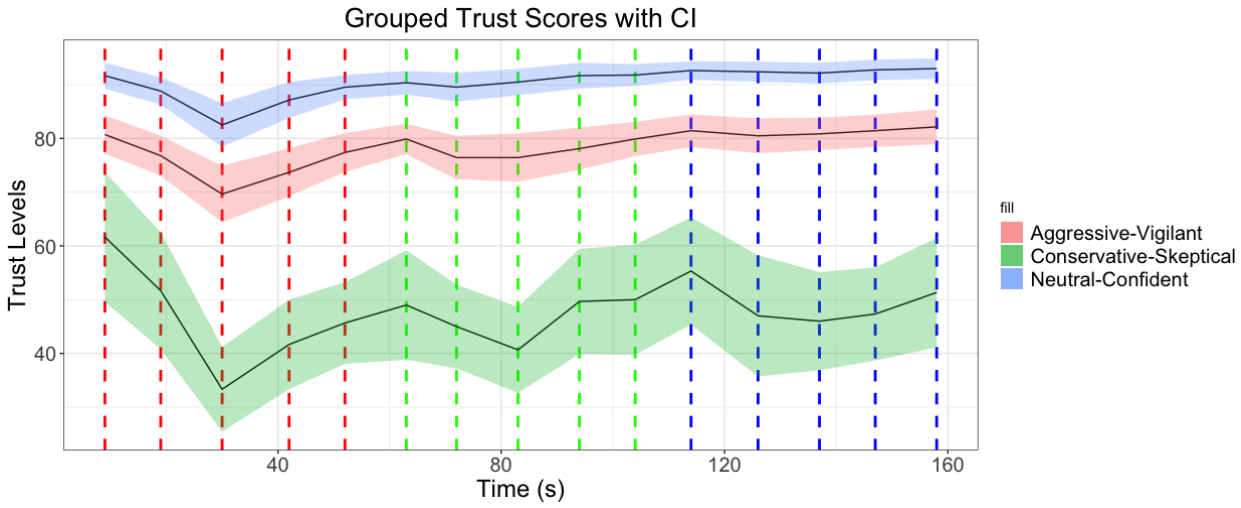


Figure 4.4: Timeline of lane change behavior (mean and 90% CI) given trust levels for three cluster groups: agressive, conservative, and neutral

4.5.2 Trust-calibrated Customized Systems

In the previous section, we build customized trust prediction models based on trust-dynamics and self-reported driving styles. Before we can integrate the trust models into the IRL framework, there are three important parameters and function (w_1, w_2, and $r'(\cdot)$) to be identified first. Considering the limited number of interactions we collected and the off-line nature of the framework, we only assign a small value of w_1 to incorporate the trust levels as additional reward. Otherwise, the optimal policies will mainly depend on the trust scores and potentially largely deviate from the base policy that we plan to fine tweak. Therefore, we assign $w_1 = 0.05$ and $w_2 = 0.95$. Additionally, $r'(\dot{})$ is more difficult to identify.

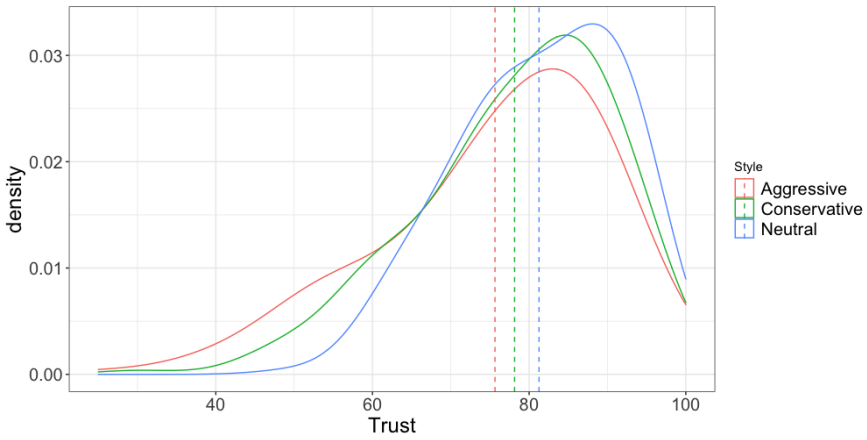
Before we dive into the specific form of $r'(\cdot)$ function, we can take a look at each cluster of drivers' trust distribution across different driving style systems. Fig. 4.5 show the trust distributions of three customized lane change systems of the three cluster of drivers. Additionally, we run Analysis of Variance (ANOVA) to test if the drivers have specific preference of a certain type of customized systems. Moreover, additional Linear Models (LMs) are used to identify the most preferred customized system if the ANOVA result rejects

Table 4.1: MSE and F1 scores for general trust model and customized trust models using different clustering criteria.

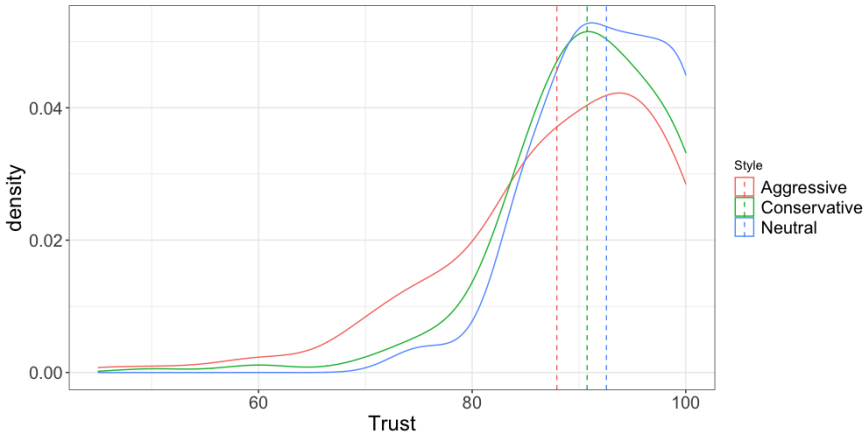
Clustering Criteria	Cluster	Number of participants	MSE for Trust	F1 score for Take-over
-	General model	99	0.832	0.326
Self-reported	"Aggressive-Vigilant"	42	0.310	0.698
Driving Style	"Neutral-Confident"	42	0.244	0.715
+Trust Dynamics	"Conservative-Skeptical"	15	0.672	0.430
	"(Very) Aggressive"	16	0.647	0.559
Driving Style	"Neutral"	24	0.513	0.493
	"(Very) Conservative"	59	0.488	0.490

the null hypothesis. From the ANOVA and LM results, the "Aggressive-Vigilant" drivers shows no significant difference in trust distributions between the three customized systems ($F(2,627)=2.78$, $p=0.06$). Similarly, the "Conservative-Skeptical" drivers also show no significant difference between the three customized systems ($F(2,222)=0.439$, $p=0.646$). However, the conservative system has the most centered density plot which has shorter tails. Lastly, the "Neutral-Confident" drivers show significant difference between the three customized systems ($F(2,627)=16.92$, p¡0.001). They prefer the neutral lane change systems the most.

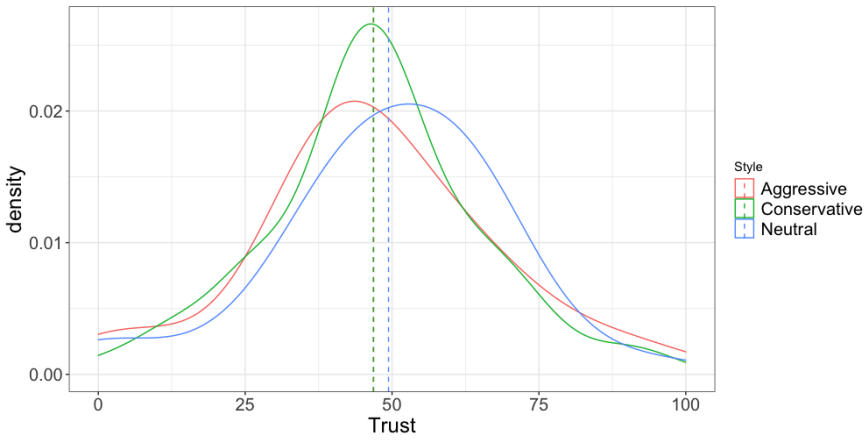
We also plot the density plot for all participants in Fig. 4.6. It shows almost identical trend compared to the "Aggressive-Vigilant" cluster plot in Fig. 4.5(a). Therefore, our sample participants demonstrates a more vigilant style regarding the customized systems. This finding matches a lot of studies in trust studies [27, 61, 87] that the drivers should stay vigilant to be ready to take-over control when there is an automation failure. Overtrusting is the major reason for overreliance which an operator's trust in automation exceeds it's actual capabilities [61]. On the contrary, undertrusting may result in the automation being ignored and negating associated benifits with its use [89]. Therefore, trust should be calibrated to an appropriate level to ensure the users trust the system but stay vigilant during interaction. With this design idea, we have the perfect data-driven trust reward function $r'(\cdot)$ that can



(a) Trust levels of "Aggressive-Vigilant" drivers.



(b) Trust levels of "Neutral-Confident" drivers.



(c) Trust levels of "Conservative-Skeptical" drivers.

Figure 4.5: Density plots for trust levels of different cluster of participants

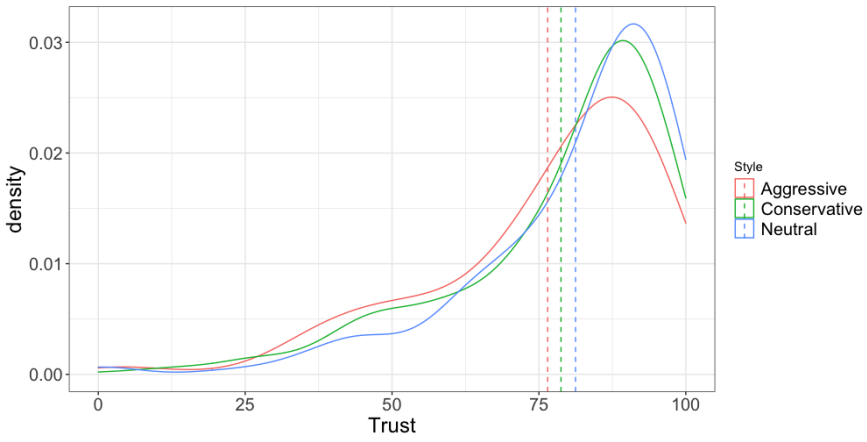


Figure 4.6: Density plot of trust levels of all participants.

be fitted from the sample distribution. Finally, since the trust distributions of the three customized systems show no significant difference between each other, we combine all data into one single distribution and fits for the trust reward function $r'(\cdot)$.

We applied Kernel Density Estimation (KDE) [108] method to estimate the probability distribution of trust levels. The non-parametric method allows relatively easy implementation and avoid the prior knowledge of generating distribution of the sample data. With KDE, we can also evaluate the probability in each data point. We selected the Gaussian kernel function and a smoothing factor of 1.5. The result is shown in Fig. 4.7. The KDE distribution has a mean of 78.82 which indicates that we will assign a larger reward value to actions that induce a trust level around 80. For trust levels that are far away from 80 in either side, the reward function will penalize the probabilities of the actions to mitigate overtrusting and undertrusting issues.

Finally, with all pieces available, we can train the IRL-SS algorithm to build trust-calibrated customized vehicle automation. To summarize all parameters used in the training step, the transition probability is obtained by taking the maximum likelihood estimate of the transition matrix [102]. However, not all states are guaranteed to be observed in the demonstrated trajectories. To account states with no observations, we adopt the common

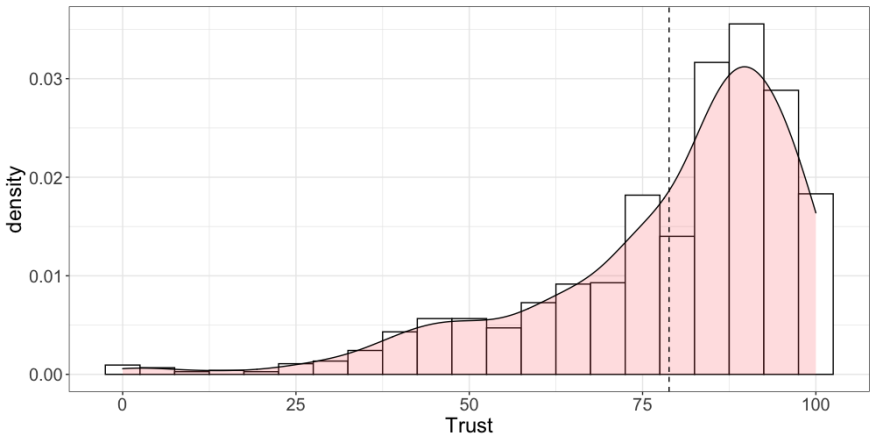


Figure 4.7: Density plot of the KDE method and histogram of the collected trust levels.

ϵ-greedy ($\epsilon = 0.1$) algorithm to get a trade-off between exploration and exploitation. The experiments are carried out using a Feed-forward Neural Network (FNN) with two hidden layers of 30 hidden units in each layer. We select the discount factor as 0.9, and the learning rate as 0.01. The reward values are normalized to a continuous scale between -1 and 1. We assign $w_1 = 0.05$ and $w_2 = 0.95$, and the trust reward function is obtained using KDE. The optimal policies are recovered using approximate value iteration in each update of the reward function.

We demonstrate the trust-calibrated stochastic policies of aggressive drivers in Tab. 4.2. When we compare Tab. 4.2 with Tab. 2.3, we notice that the optimal action in each scenario stays basically the same. Only they are reinforced to slightly higher probabilities in the stochastic policy table. This matches the observation that the current trust dynamics of the aggressive drivers are very close to the distributions of the ideal trust distribution. Therefore, the optimal actions are reinforced. Additionally, the neutral style customized systems have the highest self-reported trust levels compare to others, so Tab. 4.2 is calibrated slightly toward the neutral policy with more stay in the current lane actions and less aggressive actions. As for trust-calibrated neutral policy shown in Tab. 4.3, the policies tend to be more aggressive and take more lane change actions compare to the original neutral policy

Table 4.2: Trust-calibrated stochastic policy of the aggressive drivers for slow driving without any lead car.

Dist. to Left Vehicles	Dist. to Right Vehicles	Probability of Action		
		Stay in Current Lane	Change to Right Lane	Change to Left Lane
No info	No info	0.395	0.206	0.399
No info	Very small	0.068	0.466	0.466
No info	Small	0.324	0.233	0.433
No info	Medium	0.172	0.195	0.633
No info	Large	0.28	0.715	0.005
Very small	No info	0.729	0.147	0.124
Very small	Very small	0.655	0.173	0.173
Very small	Small	0.418	0.291	0.291
Very small	Medium	0.384	0.105	0.511
Very small	Large	0.075	0.462	0.462
Small	No info	0.197	0.801	0.002
Small	Very small	0.782	0	0.218
Small	Small	0.765	0.118	0.118
Small	Medium	0.598	0.201	0.201
Small	Large	0.394	0.143	0.463
Medium	No info	0.361	0.198	0.441
Medium	Very small	0.718	0.141	0.141
Medium	Small	0.987	0.007	0.007
Medium	Medium	0.847	0.076	0.076
Medium	Large	0.559	0.221	0.221
Large	No info	0.609	0.176	0.216
Large	Very small	0.443	0.278	0.278
Large	Small	0.37	0.521	0.118
Large	Medium	0.63	0.185	0.185
Large	Large	0.13	0.026	0.844

Note: Blue shade denotes the optimal action

shown in Tab. 2.4. This is partially due to the neutral drivers tend to overtrust the neutral system. Hence, more aggressive behaviors will alert the drivers to stay vigilant to avoid overreliance. Finally, the conservative policy has a few new actions of lane changes instead of staying in the current lane all the time in Tab. 4.4. The overall policy is still more conservative toward staying in the current lane comparing to other two policy tables.

Once successfully obtain the trust-calibrated policy tables, we test these policies on the

Table 4.3: Trust-calibrated stochastic policy of the neutral drivers for slow driving without any lead car.

Dist. to Left Vehicles	Dist. to Right Vehicles	Probability of Action		
		Stay in Current Lane	Change to Right Lane	Change to Left Lane
No info	No info	0.534	0.4	0.006
No info	Very small	0.407	0.275	0.318
No info	Small	0.315	0.315	0.37
No info	Medium	0.362	0.331	0.308
No info	Large	0.45	0.174	0.376
Very small	No info	0.25	0.403	0.348
Very small	Very small	0.348	0.256	0.396
Very small	Small	0.461	0.455	0.084
Very small	Medium	0.256	0.457	0.287
Very small	Large	0.274	0.579	0.147
Small	No info	0.541	0.234	0.225
Small	Very small	0.345	0.369	0.285
Small	Small	0.311	0.345	0.345
Small	Medium	0.297	0.302	0.401
Small	Large	0.31	0.298	0.393
Medium	No info	0.567	0.34	0.093
Medium	Very small	0.31	0.178	0.512
Medium	Small	0.273	0.363	0.363
Medium	Medium	0.278	0.415	0.306
Medium	Large	0.418	0.291	0.291
Large	No info	0.336	0.315	0.349
Large	Very small	0.248	0.453	0.299
Large	Small	0.205	0.185	0.61
Large	Medium	0.447	0.277	0.277
Large	Large	0.498	0.251	0.251

Note: Blue shade denotes the optimal action

same unobserved trips and compared with previously trained customized and non-customized systems. The action prediction accuracy is reported in Tab. 4.5. Our trust-calibrated customized systems outperforms all other systems including our previously designed IRL based customized systems for aggressive and neutral trips. However, there is also slightly decrease in the action prediction accuracy for unobserved conservative trips. This might relate to the already high prediction accuracy for conservative trips and the trust distribution

Table 4.4: Trust-calibrated stochastic policy of the conservative drivers for slow driving without any lead car.

Dist. to Left Vehicles	Dist. to Right Vehicles	Probability of Action		
		Stay in Current Lane	Change to Right Lane	Change to Left Lane
No info	No info	0.652	0.333	0.015
No info	Very small	0.322	0.118	0.56
No info	Small	0.15	0.325	0.525
No info	Medium	0.346	0.08	0.574
No info	Large	0.389	0.306	0.306
Very small	No info	0.989	0.005	0.005
Very small	Very small	0.569	0.216	0.216
Very small	Small	0.524	0.238	0.238
Very small	Medium	0.537	0.231	0.231
Very small	Large	0.562	0.219	0.219
Small	No info	0.998	0.001	0.001
Small	Very small	0.541	0.164	0.295
Small	Small	0.488	0.256	0.256
Small	Medium	0.627	0.186	0.186
Small	Large	0.645	0.178	0.178
Medium	No info	0.768	0.17	0.062
Medium	Very small	0.391	0.069	0.539
Medium	Small	0.284	0.358	0.358
Medium	Medium	0.483	0.259	0.259
Medium	Large	0.664	0.168	0.168
Large	No info	0.994	0.003	0.003
Large	Very small	0.812	0.094	0.094
Large	Small	0.22	0.39	0.39
Large	Medium	0.614	0.193	0.193
Large	Large	0.952	0.024	0.024

Note: Blue shade denotes the optimal action

from collected data can only result in lower reward values which has smaller influence on the trained policy.

4.6 Discussion

In this work, we take a small step toward human-in-the-loop customized systems which takes human real-time cognitive feedback as additional reward to calibrate the optimal policies

Table 4.5: Action prediction accuracy for unobserved trips.

Methods	Prediction Accuracy (%)		
	aggressive	neutral	conservative
Non-customized	40.64	65.33	60.20
Customized using expert-coded rewards	30.52	61.67	74.4
Customized using GAIL	60.75	68.42	70.07
Customized using IRL	73.82	70.52	85.40
Customized using IRL-SS	81.80	77.16	84.22

Note: The customized systems are tested for the corresponding trips with the same driving style.

learned from demonstrated trajectories in NDD using IRL. The Interactive RL algorithm has been widely studied in research community for many years. However, most previous studies consider explicit human feedback as guidance or reward signal in algorithm design. These designs do not fit our framework because trust states are inherently difficult to access and rarely provided by the users in an active manner. We fill in this gap by incorporating the real-time trust prediction models which can estimate the trust levels without interrupting or intruding.

In order to build our IRL-SS model, we first design an online driving simulator study to collect users' trust levels toward the proposed IRL-based customized systems. The trust data is analyzed and identified three trust dynamics corresponded to the participants' self-reported driving style. The "Aggressive-Vigilant", "Neutral-Confident", and "Conservative-Skeptical" clusters of participants show distinct trust dynamics while interacting with our customized vehicle automation. One side finding is the aggressive and conservative drivers show no clear preference toward three customized systems, but the neutral drivers show significant preference toward the neutral system.

In the design of IRL-SS algorithm, we identified the trust reward function using our collected data and previous literature from trust in automation research. Our trust reward

function takes overtrusting and undertrusting issues into consideration, and tries to calibrate trust in the appropriate level.

Finally, putting all pieces together, we trained three trust-calibrated customized systems to better capture the driver behaviors observed from NDD. The trust-calibrated policy tables demonstrate the effect of trust reward function on the stochastic optimal policy tables. And the action prediction accuracy results confirm the IRL-SS algorithm can better capture the drivers' behaviors in unobserved trips for aggressive and neutral drivers. However, partly due to the trust data collected for conservative drivers are mainly skeptical drivers, the trust model for conservative drivers will potentially predict more lower trust scores for the customized systems. These lower trust scores will later result in smaller trust reward values, and finally lead to limited calibration in the conservative systems.

Chapter 5
GENERAL CONCLUSIONS

This chapter summarizes the overall findings and contributions of the dissertation. Limitations and future research will also be discussed.

5.1 Overall Summary

The overall goal of this dissertation is to investigate effective human-in-the-loop method to design human-aware and trust-calibrated customized vehicle automation. To achieve the objective, the whole human-system integration task is divided into three subtasks: (1)From a system-only perspective, we leverage NDD to learn driving behaviors and build customized vehicle automation of different driving styles; (2) From a human-only perspective, we build customized real-time trust prediction model to ensure non-interrupting and non-intruding; (3) From a human-system integration perspective, we proposed IRL-SS algorithm to develop trust-calibrated customized vehicle automation.

5.1.1 Summary of Customized Vehicle Automation

This study introduces a systematic paradigm that starts with naturalistic driving data to identify the driving behaviors and styles for a customized automated lane change system. The driving behaviors are first extracted using Multivariate Functional Principal Component Analysis (MFPCA) with minimum prior expert knowledge. The driving styles are identified by clustering the extracted driving behaviors. An Inverse Reinforcement Learning (IRL) algorithm is then used to train the automated lane change system from grouped demonstrations of the identified driving styles to capture the preferences of a group of drivers with a similar driving style. The performance of the proposed customized automated lane change

system is compared to (1) a non-customized system trained on all the sample trips, (2) customized systems built on expert-coded reward functions, and (3) customized systems trained using a Generative Adversarial Imitation Learning (GAIL) algorithm. The results show that our method outperforms all the other systems with respect to the prediction accuracy of the lane change actions. Additionally, our method gains insights on the representative behaviors of different driving styles to enable customization of automated lane change systems.

5.1.2 Summary of Customized Trust Models

We present a methodology to develop customized model by clustering humans based on their trust dynamics. The clustering-based method addresses the individual differences in trust dynamics while requiring significantly less data than personalized model. We show that our clustering-based customized models not only outperform the general model based on entire population, but also outperform simple demographic factor-based customized models. Specifically, we propose that two models based on "confident" and "skeptical" group of participants, respectively, can represent the trust behavior of the population. The "confident" participants, as compared to the "skeptical" participants, have higher initial trust levels, lose trust slower when they encounter low reliability operations, and have higher trust levels during trust-repair after the low reliability operations. In summary, clustering-based customized models improve trust prediction performance for further trust calibration considerations.

5.1.3 Summary of Trust-calibrated Customized Systems

This work introduced a new framework, which combines the other two work, IRL-SS algorithm. We first designed an online driving simulator study to collect human trust while interacting with our customized systems. The trust distributions draw insights on drivers' trust dynamics, preferences toward the different customized systems, and design of trust reward functions. We identified three "driving style-trust dynamics" clusters of participants, "Aggressive-Vigilant", "Neutral-Confident", and "Conservative-Skeptical" groups. The MSE of trust prediction and F1 score of take-over intent prediction demonstrate that our proposed

clustering criteria outperforms the general model and driving style-based clustering criteria. After training the trust models, we use trust reward functions which are fitted to the trust distribution of the entire data using KDE. The non-parametric method of distribution fitting ensures minimum prior assumptions about the underlying generating distribution. Finally, we derive the gradient of the new reward functions with reward shaping method to connect IRL and SS. The IRL-SS algorithm is then tested and validated on unobserved driving trips and compared with stochastic policy tables learned by IRL. The results shows the trust-calibrated customized models outperforms the IRL-based systems for aggressive and neutral trips, and has comparable performance for conservative trips. The IRL-SS algorithm takes a step toward human-aware automation.

5.2 Contribution and Publications

In the customization of the automated lane change systems work, we adapted an IRL-based method to train automated systems based on the clustering of NDD into different driving styles. The MFPCA method extracts driving behaviors from NDD without prior expert knowledge. The clustering analysis on the MFPCA result identifies appropriate driving styles that supports the literature on traffic psychology. The customization of automated lane change systems using driving styles improved prediction accuracy of lane change actions for unobserved driving segments. A major contribution of this work is the customization of vehicle automation to a group of drivers with similar driving styles instead of one single driver. Our findings have implication in the future design of customized vehicle automated systems to promote appropriate acceptance and reliance. This work is accepted for publication in *IEEE Transactions on Vehicular Technology* [66].

In the second work of clustering human trust dynamics for customized real-time prediction, we proposed to use clustering methods to separate different trust dynamics across the sample population. We then develop customized trust models for each cluster of the population that account for broad individual differences in trust dynamics but allows model development with limited data. Specifically, we consider an interaction between a driver and

a SAE Level 2 driving automation and collect self-reports of trust throughout the interaction. We identify groups of users with critical differences in their trust dynamics using clustering based on trust evolution features. We demonstrate that the customized models based on these clusters significantly outperform the general model in predicting human trust as well as their take-over behavior. Additionally, although demographic factors have significant contributions to individual differences, we show that clustering based on trust dynamics-based features is more effective than simple demographic factor-based clustering for trust behavior prediction. Finally, we support the existence of the resulting clusters with literature from behavioral psychology. In summary, the contributions of this work are:

1. a framework to cluster humans based on their trust behavior dynamics;

2. identification of the "confident" and "skeptical" groups of users based on their trust in automation that is grounded in literature; and

3. improvement in prediction performance of human trust and take-over behavior with limited data available using the clustering-based customized models.

To the best of our knowledge, this is the first study that clusters users based on their trust dynamics, leading to an improved customized model for real-time trust prediction. This work is published in *Proceedings of 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* [65].

The last part of this work explores a new framework of integrating passive human feedback such as trust levels into the training process of the customized vehicle automation. Unlike other studies that work on Interactive RL algorithms, the human feedback used in our IRL-SS framework is the passive cognitive states. The main contribution of our IRL-SS algorithm is to incorporate the trust prediction model into the IRL algorithm to avoid interrupting and intruding during interaction with the automated systems. Moreover, the results demonstrate the capability of trust-calibration for customized automation. This work will be submitted to *IEEE Transactions on Intelligent Transportation Systems*.

5.3 Limitations and Future Research

There are limitations of the proposed customized automated lane change systems. First, the naturalistic driving data can exhibit different driving behaviors for certain scenarios that are all within the safety threshold. For example, if an aggressive driver has to merge into one lane from a four-lane road, they need to take action regardless of their preferred driving style. This limitation can generate irrational behaviors in the training set, which will affect the reward values and optimal policies. However, the highest probabilities of the reasonable behavior (stay in the current lane) in the training set can, however, maximize safety when implementing the policy in future testing of the proposed systems. Second, the naturalistic driving data is purely observational and lacks experimental control. For this reason, some state transitions or state-action pairs may not be observed in the data. This exploration deficiency compromises the model performance. Third, the computational cost of the IRL method can be huge for long trajectories. Future studies can consider more realistic and complex scenarios that better account for the continuous state and action spaces. A more robust method may also have to be considered to overcome the irrational behaviors observed in the training set. Nevertheless, in general, our approach is promising for customizing automated lane change systems to driving styles. While the data used in this study does not provide examples of all the driving situations, we can expand on this approach to learn other meaningful driving behaviors.

As for real-time trust prediction model, one limitation is that we adopted a K-means clustering method which can be sensitive to outliers such as bad data point in the dataset. In the future, we plan to explore a more robust clustering method. Furthermore, with more human subject data, we can explore more clusters with distinct trust dynamics.

Finally, one limitation of the trust-calibrated customized vehicle automation is the limited number of interactions to collect self-reported trust data. The IRL-based method requires a substantial amount of data to achieve valid performance. Although we demonstrated the effectiveness and potential of the proposed method, more human subject studies should be

conducted in the future to better validate the systems. Another limitation is the customized systems are still trained separately and independent from each other. However, the results in trust-calibrated systems peaks into the potential of connecting three systems into one framework since they tend to transform to each other with trust as additional reward. Therefore, future study should explore the connection between different customized systems. Additionally, the proposed systems also need to be validated on another driving simulator study to collect human trust while interacting with the trust-calibrated systems. Future research should consider close the loop of trust calibration to explore how to affect trust using specifically designed actions or policies.

BIBLIOGRAPHY

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[2] National Highway Traffic Safety Administration et al. Automated vehicles for safety. nhtsa, 2021.

[3] Kumar Akash, Wan-Lin Hu, Neera Jain, and Tahira Reid. A classification model for sensing human trust in machines using eeg and gsr. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(4):1–20, 2018.

[4] Kumar Akash, Wan-Lin Hu, Tahira Reid, and Neera Jain. Dynamic modeling of trust in human-machine interactions. In *American Control Conference (ACC), 2017*, pages 1542–1548. IEEE, 2017.

[5] Kumar Akash, Neera Jain, and Teruhisa Misu. Toward adaptive trust calibration for level 2 driving automation. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, pages 538–547, 2020.

[6] Kumar Akash, Griffon McMahon, Tahira Reid, and Neera Jain. Human trust-based feedback control: Dynamically varying automation transparency to optimize human-machine interactions. *IEEE Control Systems Magazine*, 40(6):98–116, 2020.

[7] Ali Alizadeh, Majid Moghadam, Yunus Bicer, Nazim Kemal Ure, Ugur Yavas, and Can Kurtulus. Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1399–1404. IEEE, 2019.

[8] Ahmad Aljaafreh, Nabeel Alshabatat, and Munaf S Najim Al-Din. Driving style recognition using fuzzy logic. In *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, pages 460–463. IEEE, 2012.

[9] Amazon. Amazon Mechanical Turk. *Amazon Mechanical Turk - Welcome*, 2005.

[10] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *Ai Magazine*, 35(4):105–120, 2014.

[11] HongIl An and Jae-il Jung. Decision-making system for lane change using deep reinforcement learning in connected and automated driving. *Electronics*, 8(5):543, 2019.

[12] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in neural information processing systems*, pages 5048–5058, 2017.

[13] Georges S Aoude, Vishnu R Desaraju, Lauren H Stephens, and Jonathan P How. Driver behavior classification at intersections and validation on large naturalistic data set. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):724–736, 2012.

[14] Riku Arakawa, Sosuke Kobayashi, Yuya Unno, Yuta Tsuboi, and Shin-ichi Maeda. Dqn-tamer: Human-in-the-loop reinforcement learning with intractable feedback. *arXiv preprint arXiv:1810.11748*, 2018.

[15] Dilip Arumugam, Jun Ki Lee, Sophie Saskin, and Michael L Littman. Deep reinforcement learning from policy-dependent human feedback. *arXiv preprint arXiv:1902.04257*, 2019.

[16] Christian Arzate Cruz and Takeo Igarashi. A survey on interactive reinforcement learning: Design principles and open challenges. In *Proceedings of the 2020 ACM designing interactive systems conference*, pages 1195–1209, 2020.

[17] Ira Assent. Clustering high dimensional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):340–350, 2012.

[18] Hebert Azevedo-Sa, Suresh Kumaar Jayaraman, Connor T Esterwood, X Jessie Yang, Lionel P Robert, and Dawn M Tilbury. Real-time estimation of drivers' trust in automated driving systems. *International Journal of Social Robotics*, pages 1–17, 2020.

[19] Asher Bender, Gabriel Agamennoni, James R Ward, Stewart Worrall, and Eduardo M Nebot. An unsupervised approach for inferring driver behavior from naturalistic driving data. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3325–3336, 2015.

[20] D Bezzina and J Sayer. Safety Pilot Model Deployment: Test Conductor Team Report, USDOT Report No. DOT HS 812 171, 2015.

[21] Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E Taylor, and Ann Nowé. Reinforcement learning from demonstration through shaping. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[22] Vadim A Butakov and Petros Ioannou. Personalized driver/vehicle lane change models for ADAS. *IEEE Transactions on Vehicular Technology*, 64(10):4422–4431, 2014.

[23] Stephen M Casner, Edwin L Hutchins, and Don Norman. The challenges of partially automated driving. *Communications of the ACM*, 59(5):70–77, 2016.

[24] Min Chen, Stefanos Nikolaidis, Harold Soh, David Hsu, and Siddhartha Srinivasa. Planning with Trust for Human-Robot Collaboration. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction - HRI '18*, pages 307–315, Chicago, IL, USA, 2018. ACM Press.

[25] Xin Chen, Yong Zhai, Chao Lu, Jianwei Gong, and Gang Wang. A learning model for personalized adaptive cruise control. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 379–384. IEEE, 2017.

[26] Zoran Constantinescu, Cristian Marinoiu, and Monica Vladoiu. Driving style analysis using data mining techniques. *International Journal of Computers Communications & Control*, 5(5):654–663, 2010.

[27] Ewart J de Visser, Marvin Cohen, Amos Freedy, and Raja Parasuraman. A design methodology for trust cue calibration in cognitive agents. In *International conference on virtual, augmented and mixed reality*, pages 251–262. Springer, 2014.

[28] Peter de Vries, Cees Midden, and Don Bouwhuis. The effects of errors on system trust, self-confidence, and the allocation of control in route planning. 58(6):719–735, June 2003.

[29] Murat Dikmen and Catherine Burns. Trust in autonomous vehicles: The case of tesla autopilot and summon. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1093–1098. IEEE, 2017.

[30] Sidney D'Mello and Jacqueline Kory. Consistent but modest: a meta-analysis on unimodal and multimodal affect detection accuracies from 30 studies. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 31–38, 2012.

[31] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[32] Kim Drnec and Jason S. Metcalfe. Paradigm Development for Identifying and Validating Indicators of Trust in Automation in the Operational Environment of Human

Automation Integration. In Dylan D. Schmorrow and Cali M. Fidopiastis, editors, *Foundations of Augmented Cognition: Neuroergonomics and Operational Neuroscience*, volume 9744, pages 157–167. Springer International Publishing, Switzerland, 2016.

[33] Salma Elmalaki, Huey-Ru Tsai, and Mani Srivastava. Sentio: Driver-in-the-loop forward collision warning using multisample reinforcement learning. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 28–40, 2018.

[34] Ehab ElSalamouny, Vladimiro Sassone, and Mogens Nielsen. HMM-based trust model. In *International Workshop on Formal Aspects in Security and Trust*, pages 21–35. Springer, Berlin, Heidelberg, 2009.

[35] Mica R Endsley. From here to autonomy: lessons learned from human–automation research. *Human factors*, 59(1):5–27, 2017.

[36] Anestis Fachantidis, Matthew E Taylor, and Ioannis Vlahavas. Learning to teach reinforcement learning agents. *Machine Learning and Knowledge Extraction*, 1(1):21–42, 2017.

[37] Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

[38] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. In *NIPS 2016 Workshop on Adversarial Training*, 2016.

[39] Francesca Fortuna and Fabrizio Maturo. K-means clustering of item characteristic curves and item information curves via functional principal component analysis. *Quality & Quantity*, 53(5):2291–2304, 2019.

[40] Zhenhai Gao, Tianjun Sun, and Hongwei Xiao. Decision-making method for vehicle longitudinal automatic driving based on reinforcement q-learning. *International Journal of Advanced Robotic Systems*, 16(3):1729881419853185, 2019.

[41] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. *Advances in neural information processing systems*, 26, 2013.

[42] Umar Zakir Abdul Hamid, Fakhrul Razi Ahmad Zakuan, Khairul Akmal Zulkepli, Muhammad Zulfaqar Azmi, Hairi Zamzuri, Mohd Azizi Abdul Rahman, and Muhammad Aizzat Zakaria. Autonomous emergency braking system with potential field risk

assessment for frontal collision mitigation. In *2017 ieee conference on systems, process and control (icspc)*, pages 71–76. IEEE, 2017.

[43] Clara Happ and Sonja Greven. Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113(522):649–659, 2018.

[44] Clara Happ-Kurz. Object-oriented software for functional data. *Journal of Statistical Software*, 93(5):1–38, 2020.

[45] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[46] Monika Hengstler, Ellen Enkel, and Selina Duelli. Applied artificial intelligence and trust—the case of autonomous vehicles and medical assistance devices. *Technological Forecasting and Social Change*, 105:105–120, 2016.

[47] Sebastian Hergeth, Lutz Lorenz, Josef F Krems, and Lars Toenert. Effects of take-over requests and cultural background on automation trust in highly automated driving. In *Proceedings of the Eighth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, pages 331–337, Salt Lake City, Utah, USA, 2015. University of Iowa.

[48] Bryan Higgs and Montasir Abbas. Segmentation and clustering of car-following behavior: Recognition of driving patterns. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):81–90, 2014.

[49] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[50] Carl-Johan Hoel, Krister Wolff, and Leo Laine. Automated speed and lane change decision making using deep reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2148–2155. IEEE, 2018.

[51] Kevin Anthony Hoff and Masooda Bashir. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human factors*, 57(3):407–434, 2015.

[52] W. Hu, K. Akash, T. Reid, and N. Jain. Computational Modeling of the Dynamics of Human Trust During Human–Machine Interactions. *IEEE Transactions on Human-Machine Systems*, pages 1–13, 2018.

[53] Chao Huang, Hailong Huang, Peng Hang, Hongbo Gao, Jingda Wu, Zhiyu Huang, and Chen Lv. Personalized trajectory planning and control of lane-change maneuvers for autonomous driving. *IEEE Transactions on Vehicular Technology*, 2021.

[54] Charles Lee Isbell, Michael Kearns, Satinder Singh, Christian R Shelton, Peter Stone, and Dave Kormann. Cobot in lambdamoo: An adaptive social statistics agent. *Autonomous Agents and Multi-Agent Systems*, 13(3):327–354, 2006.

[55] Edwin T Jaynes. Information theory and statistical mechanics. II. *Physical review*, 108(2):171, 1957.

[56] Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. Foundations for an empirically determined scale of trust in automated systems. *International journal of cognitive ergonomics*, 4(1):53–71, 2000.

[57] W Bradley Knox, Peter Stone, and Cynthia Breazeal. Training a robot via human feedback: A case study. In *International Conference on Social Robotics*, pages 460–470. Springer, 2013.

[58] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2641–2646. IEEE, 2015.

[59] Alex Kuefler, Jeremy Morton, Tim Wheeler, and Mykel Kochenderfer. Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211. IEEE, 2017.

[60] John D Lee and Kristin Kolodge. Exploring trust in self-driving vehicles through text analysis. *Human factors*, 62(2):260–277, 2020.

[61] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.

[62] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

[63] Chan-Chiao Lin, Soonil Jeon, Huei Peng, and Jang Moo Lee. Driving pattern recognition for control of hybrid electric trucks. *Vehicle System Dynamics*, 42(1-2):41–58, 2004.

[64] Jinying Lin, Zhen Ma, Randy Gomez, Keisuke Nakamura, Bo He, and Guangliang Li. A review on interactive reinforcement learning from human social feedback. *IEEE Access*, 8:120757–120765, 2020.

[65] Jundi Liu, Kumar Akash, Teruhisa Misu, and Xingwei Wu. Clustering human trust dynamics for customized real-time prediction. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1705–1712. IEEE, 2021.

[66] Jundi Liu, Linda Ng Boyle, and Ashis Banerjee. An inverse reinforcement learning approach for customizing automated lane change systems. *IEEE Transactions on Vehicular Technology*, pages 1–1, 2022.

[67] Marika Lüders, Tor W Andreassen, Simon Clatworthy, and Tore Hillestad. Innovating for trust. In *Innovating for Trust*. Edward Elgar Publishing, 2017.

[68] Clara Marina Martinez, Mira Heucke, Fei-Yue Wang, Bo Gao, and Dongpu Cao. Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):666–676, 2017.

[69] Daniel V McGehee, Mark Brewer, Chris Schwarz, Bryant Walker Smith, et al. Review of automated vehicle technology: policy and implementation implications. Technical report, Iowa. Dept. of Transportation, 2016.

[70] Joseph E Mercado, Michael A Rupp, Jessie YC Chen, Michael J Barnes, Daniel Barber, and Katelyn Procci. Intelligent agent transparency in human–agent teaming for multi-uxv management. *Human factors*, 58(3):401–415, 2016.

[71] Stephanie M Merritt and Daniel R Ilgen. Not all trust is created equal: Dispositional and history-based trust in human-automation interactions. *Human Factors*, 50(2):194–210, 2008.

[72] J. S. Metcalfe, A. R. Marathe, B. Haynes, V. J. Paul, G. M. Gremillion, K. Drnec, C. Atwater, J. R. Estepp, J. R. Lukos, E. C. Carter, and W. D. Nothwang. Building a framework to manage trust in automation. In *Micro- and Nanotechnology Sensors, Systems, and Applications IX*, volume 10194, page 101941U, May 2017.

[73] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods

for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[74] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[75] Marie Elisabeth Gaup Moe, Mozhgan Tavakolifard, and Svein J Knapskog. Learning trust in dynamic multiagent environments using HMMs. In *Proceedings of the 13th Nordic Workshop on Secure IT Systems (NordSec 2008)*, 2008.

[76] Neville Moray, Toshiyuki Inagaki, and Makoto Itoh. Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1):44–58, 2000.

[77] Sara Moridpour, Geoff Rose, and Majid Sarvi. Effect of surrounding traffic characteristics on lane changing behavior. *Journal of Transportation Engineering*, 136(11):973–985, 2010.

[78] Lia Morra, Fabrizio Lamberti, F Gabriele Pratticó, Salvatore La Rosa, and Paolo Montuschi. Building trust in autonomous vehicles: role of virtual reality driving simulators in hmi design. *IEEE Transactions on Vehicular Technology*, 68(10):9438–9450, 2019.

[79] Ladan Mozaffari, Ahmad Mozaffari, and Nasser L Azad. Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: A case study on san francisco urban roads. *Engineering science and technology, an international journal*, 18(2):150–162, 2015.

[80] Bonnie M Muir. Trust in automation: Part i. theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics*, 37(11):1905–1922, 1994.

[81] Bonnie M Muir and Neville Moray. Trust in automation. Part II. Experimental studies of trust and human intervention in a process control simulation. *Ergonomics*, 39(3):429–460, 1996.

[82] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE, 2018.

[83] Ben Nassi, Dudi Nassi, Raz Ben-Netanel, Yisroel Mirsky, Oleg Drokin, and Yuval Elovici. Phantom of the adas: Phantom attacks on driver-assistance systems. *Cryptology ePrint Archive*, 2020.

[84] Dario Nava, Giulio Panzani, Pierluigi Zampieri, and Sergio M Savaresi. A personalized adaptive cruise control driving style characterization based on a learning approach. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2901–2906. IEEE, 2019.

[85] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287, 1999.

[86] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *ICML*, volume 1, page 2, 2000.

[87] Dongfang Niu, Jacques Terken, and Berry Eggen. Anthropomorphizing information to enhance trust in autonomous vehicles. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 28(6):352–359, 2018.

[88] Brittany E Noah and Bruce N Walker. Trust calibration through reliability displays in automated vehicles. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 361–362, 2017.

[89] Raja Parasuraman and Victor Riley. Humans and automation: Use, misuse, disuse, abuse. *Human factors*, 39(2):230–253, 1997.

[90] Kaspar Raats, Vaike Fors, and Sarah Pink. Understanding trust in automated vehicles. In *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*, pages 352–358, 2019.

[91] James O Ramsay and Bernard W Silverman. *Applied functional data analysis: methods and case studies*. Springer, 2007.

[92] JO Ramsay. Functional data analysis: Wiley online library. 2006.

[93] Julian B. Rotter. A new scale for the measurement of interpersonal trust. *Journal of Personality*, 35(4):651–665, 1967.

[94] Julian B. Rotter. Generalized expectancies for interpersonal trust. *American Psychologist*, 26(5):443–452, 1971.

[95] Julian B. Rotter. Interpersonal trust, trustworthiness, and gullibility. *American Psychologist*, 35(1):1–7, 1980.

[96] Peter AM Ruijten, Jacques Terken, and Sanjeev N Chandramouli. Enhancing trust in autonomous vehicles through intelligent user interfaces that mimic human behavior. *Multimodal Technologies and Interaction*, 2(4):62, 2018.

[97] SAE. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, 2016.

[98] Fridulv Sagberg, Selpi, Giulio Francesco Bianchi Piccinini, and Johan Engström. A review of research on driving styles and road safety. *Human Factors*, 57(7):1248–1275, 2015.

[99] Kristin E Schaefer, Jessie YC Chen, James L Szalma, and Peter A Hancock. A meta-analysis of factors influencing the development of trust in automation: Implications for understanding autonomy in future systems. *Human factors*, 58(3):377–400, 2016.

[100] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[101] Shili Sheng, Erfan Pakdamanian, Kyungtae Han, BaekGyu Kim, Prashant Tiwari, Inki Kim, and Lu Feng. A case study of trust on autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 4368–4373. IEEE, 2019.

[102] Chris Sherlaw-Johnson, Steve Gallivan, and Jim Burridge. Estimating a markov transition matrix from observational data. *Journal of the Operational Research Society*, 46(3):405–410, 1995.

[103] Bin Shi, Li Xu, Jie Hu, Yun Tang, Hong Jiang, Wuqiang Meng, and Hui Liu. Evaluating driving styles by normalizing driving behavior based on personalized driver modeling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(12):1502–1508, 2015.

[104] Klaus Sommer. Continental mobility study 2011. *Continental AG*, pages 19–22, 2013.

[105] Malin Sundbom, Paolo Falcone, and Jonas Sjöberg. Online driver behavior classification using probabilistic arx models. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1107–1112. IEEE, 2013.

[106] Yaoyuan V Tan, Michael R Elliott, and Carol AC Flannagan. Development of a real-time prediction model of driver behavior at intersections using kinematic time series data. *Accident Analysis & Prevention*, 106:428–436, 2017.

[107] The Theano Development Team, Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

[108] George R Terrell and David W Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

[109] Richard Tomsett, Alun Preece, Dave Braines, Federico Cerutti, Supriyo Chakraborty, Mani Srivastava, Gavin Pearson, and Lance Kaplan. Rapid trust calibration through interpretable and uncertainty-aware ai. *Patterns*, 1(4):100049, 2020.

[110] Alexander Trende, Daniela Gräfing, and Lars Weber. Personalized user profiles for autonomous vehicles. In *Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications: Adjunct Proceedings*, pages 287–291, 2019.

[111] Charlott Vallon, Ziya Ercan, Ashwin Carvalho, and Francesco Borrelli. A machine learning approach for personalized autonomous lane change initiation and control. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1590–1595. IEEE, 2017.

[112] Minh Van Ly, Sujitha Martin, and Mohan M Trivedi. Driver classification and driving style recognition using inertial sensors. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 1040–1045. IEEE, 2013.

[113] Abdul Wahab, Chai Quek, Chin Keong Tan, and Kazuya Takeda. Driving profile modeling and recognition based on soft computing approach. *IEEE Transactions on Neural Networks*, 20(4):563–582, 2009.

[114] Bo Wang. *Modeling drivers' naturalistic driving behavior on rural two-lane curves.* PhD thesis, Iowa State University, 2015.

[115] Dajun Wang, Xin Pei, Li Li, and Danya Yao. Risky driver recognition based on vehicle speed time series. *IEEE Transactions on Human-Machine Systems*, 48(1):63–71, 2017.

[116] Wenshuo Wang, Junqiang Xi, Alexandre Chong, and Lin Li. Driving style classification using a semisupervised support vector machine. *IEEE Transactions on Human-Machine Systems*, 47(5):650–660, 2017.

[117] Wenshuo Wang, Junqiang Xi, and Ding Zhao. Driving style analysis using primitive driving patterns with bayesian nonparametric approaches. *IEEE Transactions on Intelligent Transportation Systems*, 20(8):2986–2998, 2018.

[118] Wenshuo Wang, Ding Zhao, Wei Han, and Junqiang Xi. A learning-based approach for lane departure warning systems with a personalized driver model. *IEEE Transactions on Vehicular Technology*, 67(10):9145–9157, 2018.

[119] Philipp Wintersberger, Anna-Katharina Frison, Andreas Riener, and Linda Ng Boyle. Towards a personalized trust model for highly automated driving. *Mensch und Computer 2016–Workshopband*, 2016.

[120] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[121] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

[122] Anqi Xu and Gregory Dudek. OPTIMo: Online Probabilistic Trust Inference Model for Asymmetric Human-Robot Collaborations. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '15, pages 221–228, New York, NY, USA, 2015. ACM.

[123] X Jessie Yang, Vaibhav V Unhelkar, Kevin Li, and Julie A Shah. Evaluating effects of user experience and system transparency on trust in automation. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, pages 408–416. IEEE, 2017.

[124] Changxi You, Jianbo Lu, Dimitar Filev, and Panagiotis Tsiotras. Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robotics and Autonomous Systems*, 114:1–18, 2019.

[125] Chao Yu, Tianpei Yang, Wenxuan Zhu, Guangliang Li, et al. Learning shaping strategies in human-in-the-loop interactive reinforcement learning. *arXiv preprint arXiv:1811.04272*, 2018.

[126] Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 295–305, 2020.

[127] Bing Zhu, Shude Yan, Jian Zhao, and Weiwen Deng. Personalized lane-change assistance system with driver behavior identification. *IEEE Transactions on Vehicular Technology*, 67(11):10293–10306, 2018.

[128] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

Appendix A

CUSTOMIZATION IN VEHICLE AUTOMATION

A.1 Driving Style Identification

Additional plots to better visualize and interpret the behaviors revealed by the functional PCs. The explicit forms of the functional PCs are in Fig. A.1 and Fig. A.3, and the effects of the functional PCs on the average speed are shown in Fig. A.2 and Fig. 2.4.

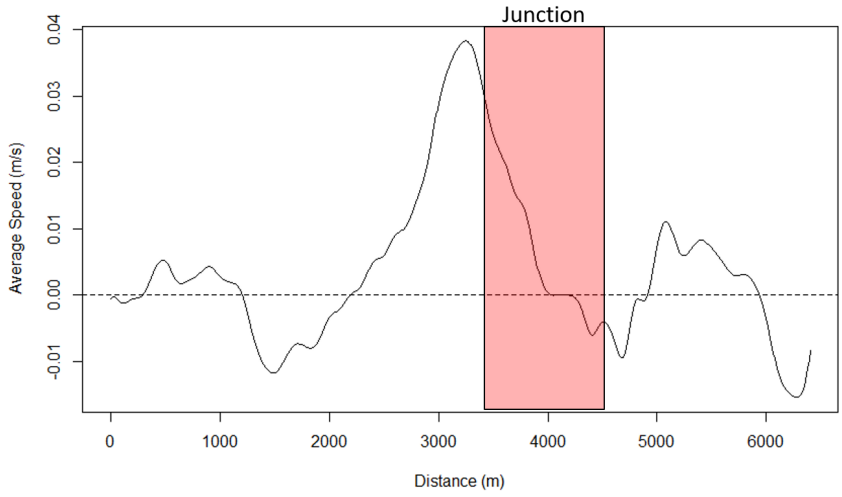


Figure A.1: Functional PC 1. Most variability is observed when entering the junction segment. A large positive coefficient denotes higher speed and faster deceleration behavior.

A.2 Customized Vehicle Automation

The detailed discretization intervals of the four continuous variables, which define the state of the automated lane change system are reported in Tab. A.1. The policy tables for aggressive

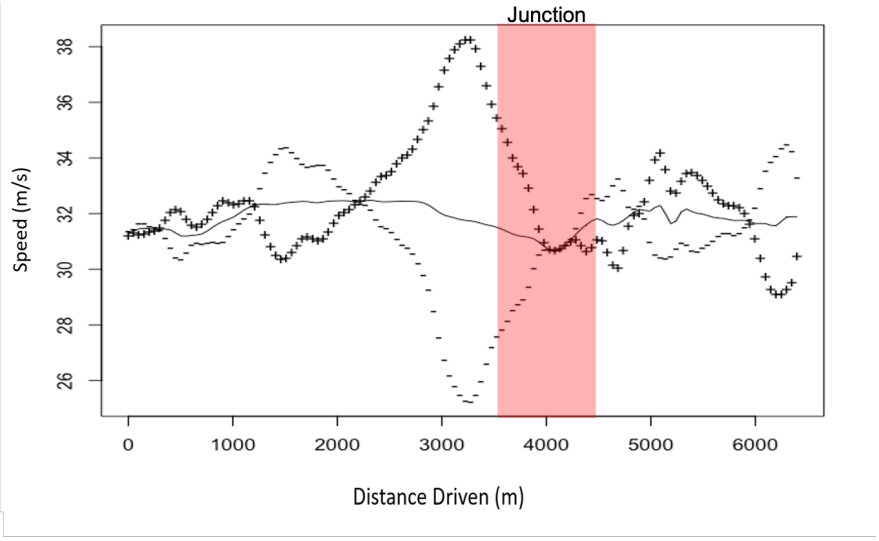


Figure A.2: The effects of the different functional PC 1 coefficients on the average speed profile. The "+" and "-" lines are generated by adding and subtracting two standard deviation times the functional PC 1 from the average speed profile.

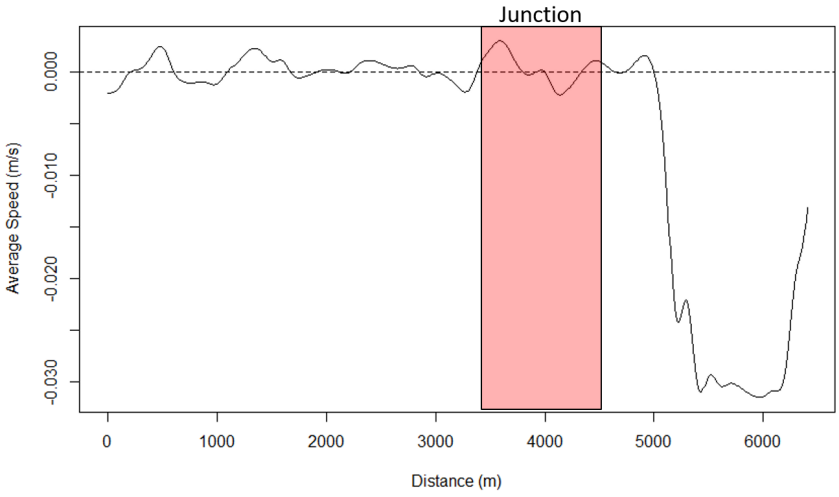


Figure A.3: Functional PC 2. Most variability is observed on the open road after the junction. A large positive coefficient denotes lower speed and faster deceleration behavior.

and neutral drivers learned from expert-coded reward functions are shown in Tables A.2 and A.3.

Table A.1: Numerical intervals for discretization of the continuous variables: speed, forward distance, left distance, and right distance.

Var. Names	Speed (m/s)	Forward Dist. (m)	Left Dist. (m)	Right Dist. (m)
Intervals		NA	NA	NA
	< 32.23	< 25.13	< 43.18	< 34.98
	$[32.23, 33.55)$	$[25.13, 39.94)$	$[43.18, 65.58)$	$[34.98, 70.52)$
	$[33.55, 35.04)$	$[39.94, 62.07)$	$[65.58, 93.93)$	$[70.52, 111.48)$
	$[35.04, 39.74]$	$[62.07, 130.19]$	$[93.93, 200.02]$	$[111.48, 200.02]$

Note: The distance is measured from the front radar system.

Last, we visualize one representative driving trip of the aggressive drivers in the test set to highlight the difference in performance of our proposed method from the expert-coded reward function. Note that this trajectory is not a driving trajectory; instead, it is the decision making sequence of the lane change actions. For Figures A.4 and A.5, the driving direction is from left to right, and the two colored trajectories in each figure show the annotation of the lane change decisions in the three lanes. Additionally, we offset the two trajectories in each plot for better visualization. We observe that the trajectory for the proposed IRL method corresponds closely to that of the actual trajectory, whereas the trajectory for the expert-coded reward function deviates from the actual trajectory at several places. This observation reinforces that our proposed method has better prediction accuracy than the expert-coded reward function.

Table A.2: Stochastic policy of the aggressive drivers using the expert-coded reward function for slow driving without any lead car.

Dist. to Left Vehicles	Dist. to Right Vehicles	Actions		
		Stay in Current Lane	Change to Right Lane	Change to Left Lane
No info	No info	0.177	0.196	0.628
No info	Very small	0.048	0.476	0.476
No info	Small	0.286	0.357	0.357
No info	Medium	0.488	0.456	0.056
No info	Large	0.333	0.009	0.658
Very small	No info	0.515	0.056	0.43
Very small	Very small	0.002	0.499	0.499
Very small	Small	0.009	0.496	0.496
Very small	Medium	0.2	0.069	0.732
Very small	Large	0.013	0.493	0.493
Small	No info	0.219	0.387	0.394
Small	Very small	0.023	0	0.977
Small	Small	0.332	0.334	0.334
Small	Medium	0.486	0.257	0.257
Small	Large	0.03	0.001	0.969
Medium	No info	0.092	0.633	0.275
Medium	Very small	0.01	0.495	0.495
Medium	Small	0.976	0.012	0.012
Medium	Medium	0.838	0.081	0.081
Medium	Large	0.857	0.072	0.072
Large	No info	0.298	0.197	0.504
Large	Very small	0.03	0.485	0.485
Large	Small	0.294	0.086	0.619
Large	Medium	0.811	0.094	0.094
Large	Large	0.462	0.024	0.514

Table A.3: Stochastic policy of the neutral drivers using the expert-coded reward function for slow driving without any lead car.

Dist. to Left Vehicles	Dist. to Right Vehicles	Actions		
		Stay in Current Lane	Change to Right Lane	Change to Left Lane
No info	No info	0.924	0.073	0.003
No info	Very small	0.476	0.321	0.203
No info	Small	0.388	0.002	0.61
No info	Medium	0.246	0.262	0.492
No info	Large	0.838	0.161	0.001
Very small	No info	0.165	0.199	0.636
Very small	Very small	0.146	0	0.854
Very small	Small	0	0.001	0.998
Very small	Medium	0	1	0
Very small	Large	0	1	0
Small	No info	0.971	0	0.029
Small	Very small	0	1	0
Small	Small	0.601	0.199	0.199
Small	Medium	0.859	0.058	0.082
Small	Large	0.982	0.005	0.013
Medium	No info	0.412	0.588	0
Medium	Very small	0.004	0	0.996
Medium	Small	0.97	0.015	0.015
Medium	Medium	0.998	0	0.002
Medium	Large	0.992	0.004	0.004
Large	No info	0.579	0.421	0.001
Large	Very small	0.014	0.985	0.001
Large	Small	0.93	0.013	0.057
Large	Medium	0.995	0.003	0.003
Large	Large	0.997	0.002	0.002

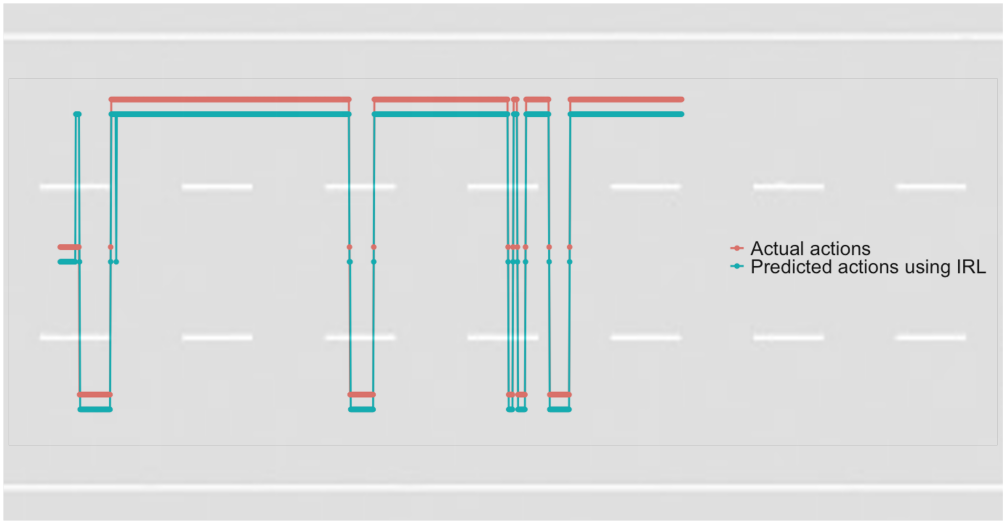


Figure A.4: Comparison of a representative trajectory between an actual aggressive trip and the corresponding trip predicted by our proposed IRL method.

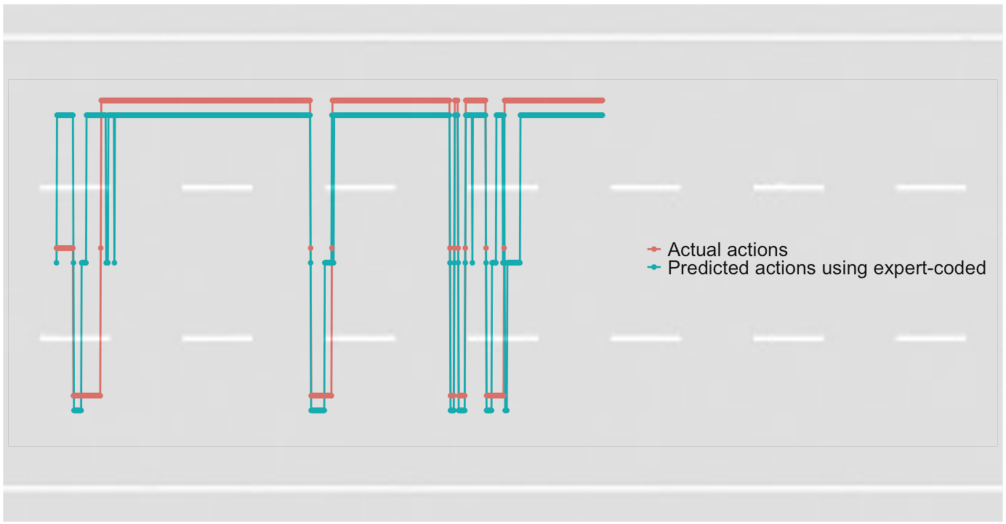


Figure A.5: Comparison of a representative trajectory between an actual aggressive trip and the corresponding trip predicted by the expert-coded reward function.