2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

A Study on Learning and Simulating Personalized Car-Following Driving Style

Shili Sheng[1], Erfan Pakdamanian[1], Kyungtae Han[2], Ziran Wang[2], and Lu Feng[1]

***Abstract*— Automated vehicles are gradually entering people's daily life to provide a comfortable driving experience for the users. The generic and user-agnostic automated vehicles have limited ability to accommodate the different driving styles of different users. This limitation not only impacts users' satisfaction but also causes safety concerns. Learning from user demonstrations can provide direct insights regarding users' driving preferences. However, it is difficult to understand a driver's preference with limited data. In this study, we use a model-free inverse reinforcement learning method to study drivers' characteristics in the car-following scenario from a naturalistic driving dataset, and show this method is capable of representing users' preferences with reward functions. In order to predict the driving styles for drivers with limited data, we apply Gaussian Mixture Models and compute the similarity of a specific driver to the clusters of drivers. We design a personalized adaptive cruise control (P-ACC) system through a partially observable Markov decision process (POMDP) model. The reward function with the model to mimic drivers' driving style is integrated, with a constraint on the relative distance to ensure driving safety. Prediction of the driving styles achieves 85.7% accuracy with the data of less than 10 car-following events. The model-based experimental driving trajectories demonstrate that the P-ACC system can provide a personalized driving experience.**

I. Introduction

Automated vehicles have attracted significant attention from the research community because of their promising ability to increase traffic safety and enhance the human driving experience for various driving situations. Most automated driving technologies are currently at SAE level 2 [1] relying on the development of advanced driver-assistance systems (ADAS) [2]. Car-following is a fundamental building block of automated longitudinal motion control, various ADAS are further developed to enable a safer driving experience, including cruise control (ACC) and forward-collision warning [3].

Most available ADAS are generic and user-agnostic, which limits their ability to fit drivers' different styles while different drivers tend to have different driving styles [4]. For example, an ACC can be too aggressive for a driver who prefers a large distance gap to preceding vehicles, while too conservative for a driver who prefers a smaller gap [5]. Personalized ADAS aims to integrate drivers' preferences into the system design and adapt to diverse drivers' styles [6].

[1]Shili Sheng, Erfan Pakdamanian, and Lu Feng are with School of Engineering and Applied Science, University of Virginia, Charlottesville, VA 22904, USA {ss7dr, ep2ca, lf9u}@virginia.edu

[2]Kyungtae Han and Ziran Wang are with Toyota Motor North America-InfoTech Labs, Mountain View, CA 94043, USA {kyungtae.han, ziran.wang}@toyota.com

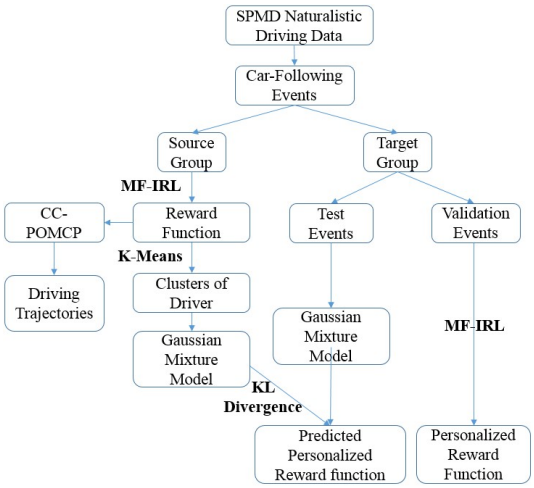

Fig. 1. Overview of the proposed method.

Because of the system ability to predict and mimic driving behavior, it can provide optimal driving experience, improve traffic safety [7], and enhance drivers' trust [8].

Learning the individual driving styles allows to predict drivers' maneuvers, make a decision at the individual level, and facilitate the design of personalized ADAS. Driving preferences can be learn through simply demonstrating driver's driving style [9], rather than manually tuning various parameters to adapt to drivers' behaviors [10] since this can be considered that drivers often maximize some reward in their mind and trade-off different factors, such as speed, acceleration, and distance to surrounding vehicles in many Inverse Reinforcement Learning (IRL) setting [11].

IRL was commonly used to infer the driving styles by learning the cost function that best explains the observed demonstrations [12, 13]. A cost function usually combines various features and weights [14]. Using cost functions to represent driving styles can also facilitate predicting and imitating driver behaviors, and further generate comfortable motions and predictable behaviors [15]. Currently, most IRL methods rely on the prior knowledge of the transition model, and it is often difficult to satisfy in real life [16]. Model-free IRL methods relax this requirement [17], and it can achieve a better performance than traditional model-based IRL methods [18].

In real-life situations, driving observations from one specific driver may be insufficient to develop individual-based personalized ADAS [19, 20, 21]. Group-based personalization studies drivers' behavior with a small number of

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922548

representative styles [21, 22]. The challenge lies in how to adapt group-based personalization to individual-based personalization when only limited data from a specific driver is available.

In this paper, we present a personalized car-following style learning method for drivers with limited data as shown in Figure 1. We used a model-free IRL to learn each individual driver's style (i.e., reward function) from 42 drivers and cluster them into four different groups. Furthermore, we developed Gaussian Mixture Models (GMM), used the Kullback–Leibler (KL)-Divergence to measure the similarities, and predict the personalized reward function. In order to validate this approach, we developed a car-following model with a partially observable Markov decision process (POMDP) to mimic drivers' driving preferences with a constraint on the minimum distance between the vehicle to ensure safety.

We highlight the following contributions in this work:

- The effectiveness of the model-free IRL in learning driving preference from naturalistic driving data is demonstrated.
- Driving preferences are clustered into four representative styles based K-means.
- Personalized driving styles are learned for drivers with limited data based on the KL-Divergence of GMMs.
- We designed a personalized ACC control utilizing the learned IRL reward function and POMDP with a constraint on the minimum distance between the vehicle to ensure safety.

The remainder of the paper is organized as follows: Section II summarizes the related work, Section III shows an overview of our method, Section IV describes the dataset and the preprocessing, Section V introduces the car-following preference learning based on the IRL, Section VI explains the driving styles clustering, Section VII presents the prediction using drivers' limited information, Section VIII illustrates the design of the POMCP model for the personalized ACC controller, and Section IX draws the conclusion.

II. Related work

A. Personalized Advanced Driver-Assistance Systems

Various personalized ADAS are proposed to assist the car-following scenarios. Group-based personalized ACC has been developed by assigning drivers into a small number of groups [21]. Gao et al. proposed a personalized ACC based on three driving styles learned from 66 drivers by using unsupervised clustering methods, and then the parameters representing each style are fed into a model predictive controller (MPC) [23]. Zhu et al. used KL-divergence to measure the similarity among 84 drivers and classified them into three groups [24]. A personalized ACC was further designed to meet different groups' driving characteristics in speed control and distance control. Wang et al. developed a learning-based personalized driver model based on bounded generalized GMM for car-following scenarios [25]. Chen et al. [26] proposed a learning model for personalized ACC that can learn and replicate driver behaviors within acceptable errors. Wang et al. developed a Gaussian Process Regression algorithm for personalized ACC, where both numerical and human-in-the-loop experiments verify the effectiveness of the proposed algorithm in terms of reducing the interference frequency by the driver [27] .

B. Inverse Reinforcement Learning

Ng and Russell [28] introduced an IRL to extract a reward function for observed optimal behaviors. Abbeel and Ng [11] used an IRL to learn five different driving styles on a highway simulation, assuming the reward function can be expressed as a linear combination of known features. Kuderer et al. [14] introduced a feature-based IRL method for learning individual navigation styles from the real driving demonstrations on a highway. They demonstrated that their method was able to achieve distinct mean acceleration and jerk for two users. They concluded that distinct policies can be learned for different users. [29] learned different rewards for three different driving styles based on an IRL method using locally optimal demonstration. Naumann et al. [15] investigated the suitability of explaining human driving behavior with the cost functions. They explored various features such as longitudinal acceleration and longitudinal jerks from human driver trajectories. They inferred human drivers' preference over the features through the learned cost function weights. Jain et al. [30] proposed a model-free IRL using maximum likelihood estimation to investigate drivers' preferences in the scenario of freeway merging based on 12 trajectories. The studies by Gao et al. [31] and Zhao et al. [32] both show that their IRL algorithms can recover the personalized car-following gap preference based on different vehicle speed values, where a gap-speed matrix can be used to design the control logic for personalized ACC systems.

III. Research Overview

In Figure 1, we summarized our method for learning the personalized car-following preference for drivers with limited data. We extracted car-following events from a naturalistic driving dataset. We kept the 49 drivers with more than 10 car-following events. We considered 42 drivers with no more than 60 events as the *source group* and the seven drivers as the *target group*.

For the drivers in the source group, we employed a model-free IRL method to infer their driving styles through the reward functions. We further clustered these driving styles into four representative styles using K-Means.

In order to demonstrate how to infer the driving style of a driver with limited data, we separated 10 events as the *test events* for each driver in the target group, and keep the rest as the *validation events*. Assuming we would not be able to learn the driving styles through the IRL directly with the limited data in the test events, we developed a GMM for them individually instead. Then we computed the individual diver's similarity with the GMM models of the 4 clusters using KL-divergence, respectively. We inferred which cluster that the driver is most similar with according to the KL-divergence. Then we assigned that individual driver with the

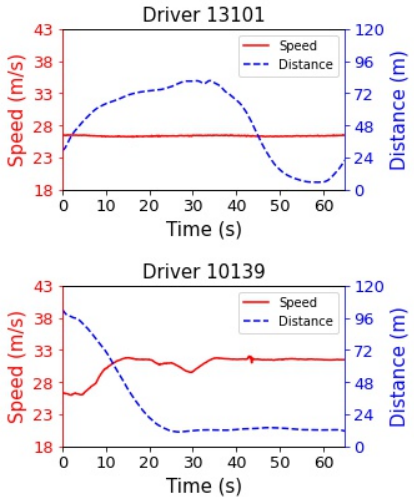

Fig. 2. Car-Following events.

centroid reward function of that cluster. Thus, we were able to predict the personalized reward function with limited data. To validate the effectiveness of this method, we used the validation events to learn the reward function directly through the IRL as the ground truth. We demonstrated the prediction accuracy in Section VII.

IV. Dataset

We extracted car-following data from the datasets of *DataWsu* and *DataFrontTargets* in the project of Safety Pilot Model Deployment (SPMD)[1]. The SPMD project collected naturalistic driving data (i.e without any restriction of particular routes or driving time) in Ann Arbor, Michigan, USA. Ninety-eight sedans integrated with data acquisition systems (DAS) were provided to drivers. *DataWsu* recorded data such as speed and acceleration via the vehicles' Controller Area Network (CAN) Bus at a frequency of 10 Hz. *DataFrontTargets* recorded data such as relative distance and relative speed using the Mobileye at a frequency of 10 Hz. Each vehicle had a unique device ID (eg. 10205). We assumed that each vehicle was only assigned to one individual driver, and we used the device ID as the driver ID. We obtained the car-following data for 85 drivers with the following criteria to select car-following events.

- The ego vehicle followed its closest preceding vehicle for at least 30 seconds.
- The relative distance between the ego vehicle and preceding vehicle was always shorter than 120 meters. If the relative distance was longer than 120 meters, we assume that the vehicle was under free driving [33].
- The speed of the ego vehicle was always between 18 m/s and 43 m/s.

In total, there were 36 drivers with no more than 10 car-following events, and 49 drivers with more than 10 events. We kept the 49 drivers for further study.

[1]https://catalog.data.gov/dataset/safety-pilot-model-deployment-data

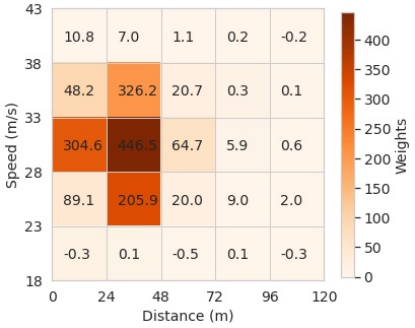

Fig. 3. Weights of the reward function for Driver 10575.

Figure 2 shows two car-following events: Driver 13101 was driving at relatively low speed (26.4 ± 0.08 m/s) while the distance to the front vehicle was changing. The driver focused on keeping the speed stable while being indifferent to the changing distance. On the contrary, Driver 10139 increased the speed to shorten the distance, and then kept at a stable speed and distance.

V. Inverse Reinforcement Learning

A. Preliminaries

The reward function in reinforcement learning determines the policy that the agent will adopt to act in an environment. However, the reward function is not always available. Instead, an expert's behavior is easier to observe. The IRL aims to derive the reward function from the observed behaviors [34]. In the car-following scenario, different drivers have their own driving styles and value the environmental factors differently. Drivers' driving styles determine their driving actions as the reward function determines the agent's policy. We can learn different drivers' driving styles using the IRL as we derive the reward function from the recorded car-following data.

The reward function is usually formed as a linear combination of binary features $\phi : S \times A \rightarrow \{0, 1\}$, where S denotes the state space that the agent can perceive in the environment, and A denotes the action space that the agent can perform. The reward function for expert E can be denoted as $R_E(s, a) = \sum_{m=1}^{M} \omega_m \cdot \phi_m(s, a)$, where M is the number of features and ω is the weight. Ultimately, the IRL is learning the weights such that the demonstrated behavior is optimal.

Model-based IRL methods such as Bayesian inference [35], maximum entropy [36], and maximum likelihood estimation [18] require prior knowledge of transition function, which is not easy to obtain in real life. Jain et al. [30] proposed a model-free IRL method *Q-averaging* by estimating the Q-value without knowledge of the transition function and it achieves a higher log-likelihood compared with an existing model-based method. We employed the *Q-averaging* IRL method in this study.

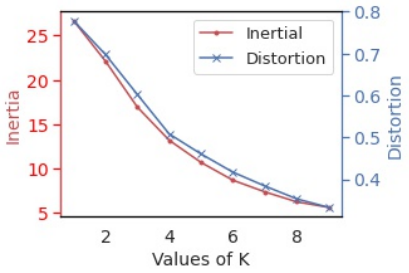

Fig. 4. Inertia and distortion with respect to the value of K.

B. Learning Car-Following Preference

In order to capture the variability of the driving states, we aggregated data of every three seconds into one data point. We summarized the setup of the car-following data with IRL format as follows.

State Space. We defined the state space with two state variables:

- d: The relative distance from the ego vehicle to its closest preceding vehicle.
- v: The velocity of the ego vehicle from the vehicle's CAN Bus.

We discretized the speed of the vehicles and their relative distance to front vehicles into five intervals evenly, which led to 25 states for each vehicle.

Action. We modeled the acceleration (in m/s^2) as the actions of the drivers.

- High brake $(acc \leq -1.46)$,
- Mild brake $(-1.46 < acc \leq -0.18)$,
- Minimal acceleration $(-0.18 < acc \leq 0.18)$,
- Mild acceleration $(0.18 < acc \leq 1.46)$,
- High acceleration $(acc > 1.46)$.

Feature. We used 25 features to indicate the states of relative distance and velocity of the vehicle.

Reward Function. Figure 3 shows the learned feature weights for Driver 10575 using the model-free IRL approach. It explains the driver's preference over different driving states. The weight of the 12^{th} feature (speed between 28 m/s and 33 m/s, distance between 24 m and 48 m) is the greatest, corresponding to the state that Driver 10575 is most comfortable with. The cumulative weights over the states with the distance between 24 m and 48 m are greater than the weights over the states with speed between 28 m/s and 33 m/s, indicating that he/she prefers maintaining a stable distance rather than maintaining a stable speed. The states with fewer weights (e.g., states with speed below 23 m/s, and states with distance above 96 m) show the driving situations that the driver prefers not to be in.

VI. Clustering

We obtained weights of the reward function for each driver using the model-free IRL method as a 25-dimensional vector. Then we normalized the vector into $[-1, 1]$ for each driver without losing the information of the driver's preference over different states. We divided the drivers into K separate groups based on the normalized weights using K-means and the elbow method.

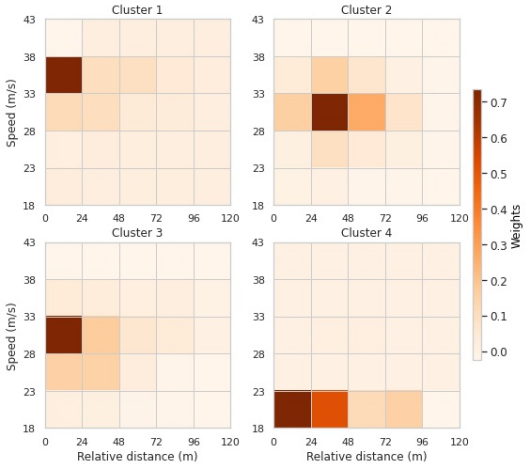

Fig. 5. Centroid of the normalized weights of the reward function for each group.

The goal of K-means clustering is to divide observations into K clusters such that each observation belongs to the clustering with the nearest mean. The elbow method is commonly used to find an optimal number of clusters.

Figure 4 shows the inertia and distortion with respect to the value of K. Inertia is the sum of squared distances of samples to their closest cluster center, and distortion is the average of the squared distances from the cluster centers of the respective clusters using the Euclidean distance metric. Both inertia and distortion will decrease with the increase of clustering number K as the sample partition becomes more refined. The decrease will be sharp before reaching the true clustering number, and it will become flat afterward [37]. As shown in Figure 4, the decrease is sharper before K reaches 4 and becomes relatively flat after reaching 4. Therefore, we selected the number of representative driver groups in the car-following scenario K as 4.

Figure 5 shows the centroid of the normalized weights for each group. Drivers in Cluster 1 prefer to drive at a relatively high speed with a short distance, representing a confident driving style. The drivers in Cluster 2 and Cluster 3 prefer the state with the speed between 28 m/s and 33 m/s, while the drivers in Cluster 2 prefer greater distance. Drivers in Cluster 4 prefer to drive at a relatively low speed and short distance, which is usually the strategy when the traffic is relatively heavy.

Figure 6 shows the histograms of the speed and the distance for different clusters of drivers. It is evident that the speed and the distance distributions for each cluster have different shapes, indicating different driving styles. For the speed histograms, Cluster 1 has a plateau between 32 m/s and 36 m/s, Cluster 2 and Cluster 3 are skewed between 28 and 33 m/s, and Cluster 4 has a peak at 26 m/s.

Figure 7 shows the boxplots of the driving indicators, including the speed and the distance for the drivers in each cluster. It also demonstrates the difference between the

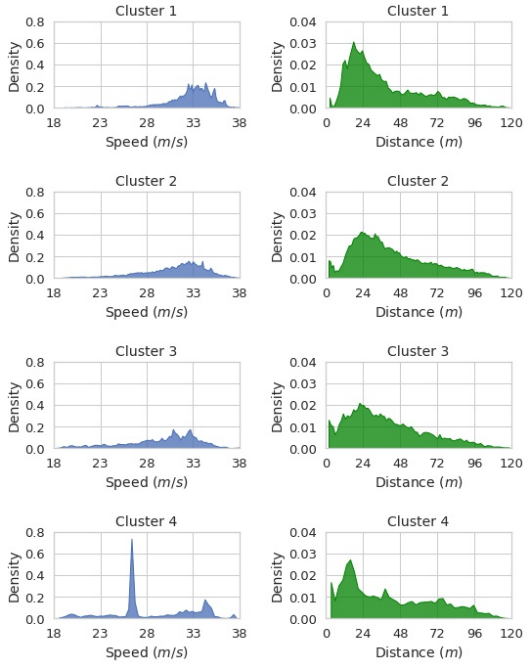

Fig. 6. Histogram of the speed and the distance for each cluster of drivers.

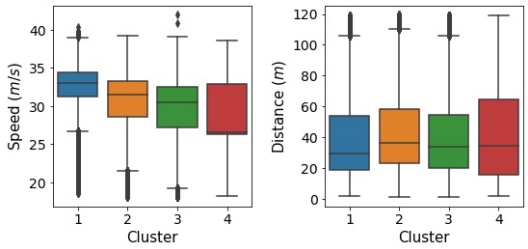

Fig. 7. Boxplot of the speed and the distance for each cluster of drivers.

drivers in each cluster. For the speed boxplots, Cluster 1 has the greatest median value and Cluster 4 has the least. Cluster 2 and cluster 3 are close. For the distance boxplots, Cluster 4 has the largest range. These characteristics are consistent with the styles represented by the reward function for each cluster.

VII. Prediction

GMM has proven its effectiveness in modelling various driving behaviors [24]. The driving data can be modelled as a linear combination of Gaussian distributions:

$$p(x) = \sum_{i=1}^{M} \pi_i p(x|\mu_i, \sigma_i), \tag{1}$$

where M is the number of Gaussian distribution, the i^{th} component is a multivariate Gaussian distribution $G(\mu_i, \sigma_i)$ with weight π_i.

We considered the drivers with more than 60 car-following events as the target set, the drivers with car-following events between 11 and 60 as the source set. We further divided the car-following events in the target set into two parts: 1) Ten events for GMM modeling; 2) The rest events for validation by applying the IRL directly.

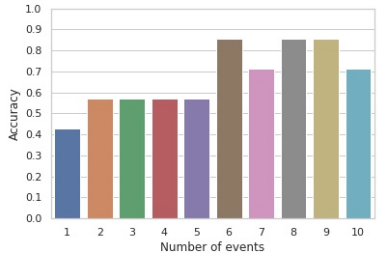

Fig. 8. Prediction accuracy.

We first modeled each driver with limited data (up to 10 events in Part 1) using a GMM $f(x)$ individually, then we modelled each cluster of drivers in the source set into a GMM $g(x)$. For each driver in the target set, we aim to find the most similar cluster of drivers. We used KL-divergence is often used to measure the similarity. [24]:

$$D(f||g) = \int f(x) \log \frac{f(x)}{g(x)} dx \tag{2}$$

Given that the integral is not tractable, the Monte-Carlo sampling method is used to approximate the KL-divergence. The smaller the KL-divergence, the greater the similarity. In other words, we aimed to find the smallest KL-divergence.

We directly learned the reward function from the events in Part 2 through the IRL, and used its closet cluster as the ground truth. Figure 8 shows the prediction accuracy. When the number of events used to build the GMM model is less than 6, the accuracy is less than 60%. When the number increased to 6 and beyond, the accuracy also increased. The best accuracy is 85.7% for the event numbers 6, 8, and 9. It demonstrates that our method has the potential to predict a personalized reward function for the drivers with limited data.

VIII. Online Controller Based on Cost-Constrained POMCP

We designed a personalized adaptive cruise controller based on cost-constrained partially observable Monte-Carlo Planner (CC-POMCP) using the learned IRL reward.

A. Preliminaries

Formally, a POMDP is denoted as a tuple $(S, A, \mathcal{T}, R, O, \delta, \gamma)$, where S is a finite of state, A is a set of actions, $\mathcal{T}$ is the transition function representing conditional transition probabilities between states, $R : S \times A \to \mathbb{R}$ is the real-valued reward function, O is a set of observations, δ is the observation function representing the conditional probabilities of observations given states and actions, and $\gamma \in [0, 1]$ is the discount factor. At each time step t, given an action $a_t \in A$, a state $s_t \in S$ evolves to $s_{t+1} \in S$ with probability $\mathcal{T}(s_{t+1}|s_t, a_t)$. The agent receives a reward $R(s_t, a_t)$, and makes an observation $o_{t+1} \in O$ about the next state s_{t+1} with probability $\delta(o_{t+1}|s_{t+1}, a_t)$. The goal

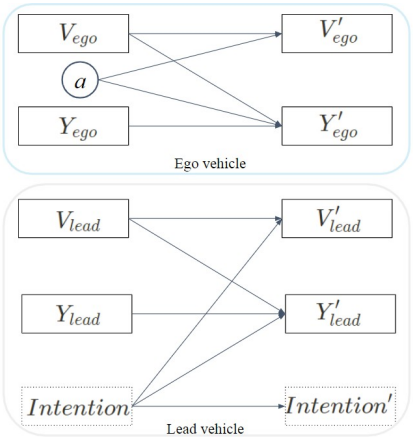

Fig. 9. Transition model.

of POMDP planning is to compute the optimal policy that chooses actions to maximize the expectation of the cumulative reward $V_R = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$. Constrained POMDP is a generalization of POMDP for multiple objectives. Its goal is to compute the optimal policy that maximizes V_R while constraining the expected cumulative costs $V_C = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t C(s_t, a_t)]$, where $C(s_t, a_t)$ below a threshold c.

B. PACC POMDP model

We designed a POMDP model as shown in Figure 9 to simulate the driving situation that one ego vehicle following a leading vehicle on a straight highway. We used v_{ego} and y_{ego} to represent the speed and the position of the ego vehicle, respectively. We also used v_{lead}, y_{lead}, and i_{lead} to represent the speed, the position of the leading vehicle. In addition, we modeled the intention of the leading vehicle's driver as a hidden state. The intention can take one of the three values, namely hesitating, normal, and aggressive. The action a in our POMDP model is the acceleration of the ego vehicle. In our experiment, the acceleration can take of the three values, namely $-0.6\ m/s^2$, 0, and $0.6\ m/s^2$. The state transition can be represented as below,

$$\begin{bmatrix} v'_{ego} \\ y'_{ego} \\ v'_{lead} \\ y'_{lead} \end{bmatrix} = \begin{bmatrix} v_{ego} \\ y_{ego} \\ v_{lead} \\ y_{lead} \end{bmatrix} + \begin{bmatrix} at \\ v_{ego}t + 0.5at^2 \\ a_{lead}t \\ v_{lead}t + 0.5a_{lead}t^2 \end{bmatrix},$$

where the time step is t.

We assume that the behavior of the leading vehicle is dictated by its driver's intention, as described in Table I. For example, if the intention of the driver in the leading vehicle is hesitating, we assume that the probability of breaking ($(a_{lead} = -0.5\ m/s^2)$) is 0.3, of maintaining ($a_{lead} = 0$) is 0.4, and of accelerating ($a_{lead} = 0.5\ m/s^2$) is 0.3.

We designed a reward function as shown in Figure 10. The reward represents drivers' driving style. In addition, we designed a cost value of 10 for the situation when the distance between the ego vehicle and leading vehicle is smaller than 2 m to ensure safety.

TABLE I
DISTRIBUTION OF ACCELERATION BASED ON A DIFFERENT INTENTION

Intention	Acceleration		
	Break	Maintain	Acceleration
Hesitating	0.3	0.4	0.3
Normal	0.1	0.8	0.1
Aggressive	0.4	0.2	0.4

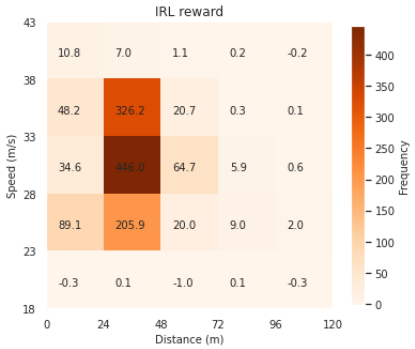

Fig. 10. The reward function for PACC.

C. CC-POMCP

We employed a CC-POMCP solver for our experiment [38]. A state is sampled from the root node's belief and then further used to sample a trajectory. More simulations tend to yield a higher cumulative reward (see Figure 11) and lower cumulative cost (see Figure 12). However, more simulations required a longer computational time. Since we aimed to output control commands at 1 Hz, we limited the computational time to be less than 1 second for each command output.

D. Results

We generate 30 car-following trajectories. In each trajectory, the ego vehicle followed the leading vehicle to the destination on a straight way. As discussed above, we set a cost constraint to ensure a safe distance. At the same time, the acceleration controlled by the CC-POMCP solver tries to mimic the driver's driving preference. Figure 14 shows the scatter plot of the speed and distance of all the 30 trajectories, and their histogram, respectively. It shows that the trajectories are mostly scattered in the states with relatively high rewards as shown in Figure 10.

As shown in Figure 13, the average trajectory of the 30 trajectories is relatively stable. The speed maintains between 28 m/s and 33 m/s, and the relative distance maintains between 24 m and 48, which is the most preferred state in Figure 10.

IX. Discussion and Conclusion

This work proposed a method to learn personalized driving styles for drivers with limited data. We first extracted car-following events from a naturalistic driving dataset, and divided them into a source set and a target set. We adopt a model-free IRL to learn the driving styles through reward function in the car-following scenarios for the events in the

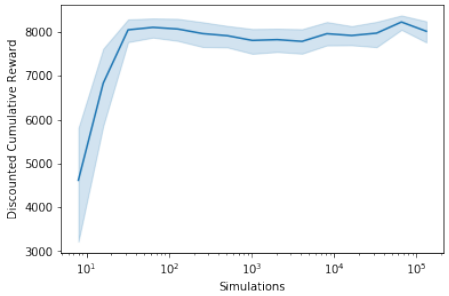

Fig. 11. The cumulative reward with respect to the number of simulations.

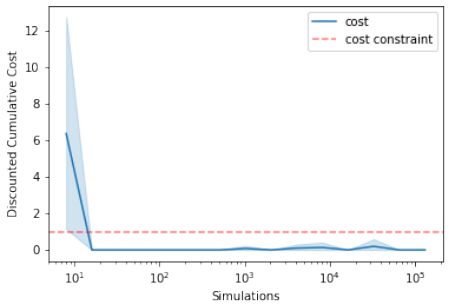

Fig. 12. The cumulative cost with respect to the number of simulations.

source set. We further clustered these driving styles into 4 representative groups. However, the IRL requires enough data and it is not always available for newly-involved drivers. We used GMM for the driver style in the target set using up to 10 events. Then we used KL-divergence to find the most similar cluster in the source set. To validate our method, we used more than 50 events in the target set to learn the reward function as the ground truth to verify whether the prediction-based KL-divergence is valid or not. We achieved a prediction accuracy of 85.7% in this study. Furthermore, we employed a CC-POMCP solver for the control of the ego vehicle in a car-following experiment. By incorporating the reward function learned from drivers, we are able to mimic drivers' driving styles and provide a personalized driving experience.

There are a few directions for future work. First, we would like to expand intervals of the states and incorporate more features. We believe it will provide a more comprehensive analysis of drivers' preferences. Another important future work is to adapt the personalized model to a more complex driving situation.

ACKNOWLEDGEMENT

This work was supported in part by National Science Foundation grants CCF-1942836, CNS-1755784, and Toyota Motor North America "Digital Twins" project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the grant sponsors.

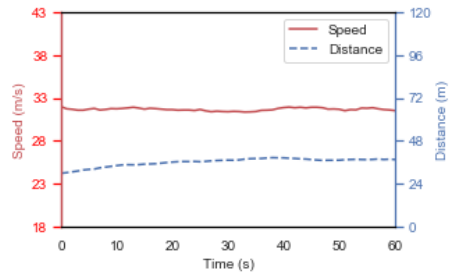

Fig. 13. The average trajectory of the ego vehicle from 30 runs of the experiment.

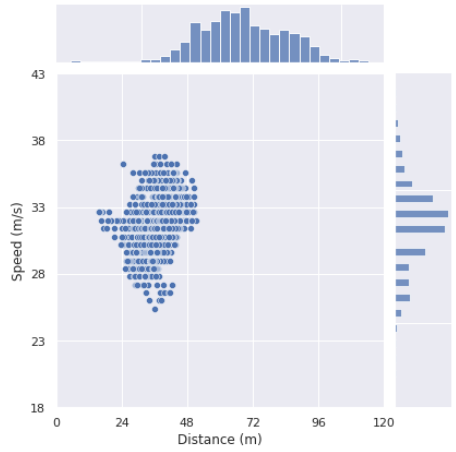

Fig. 14. The scatter plot and histogram of the speed and distance of the ego vehicle.

REFERENCES

[1] Saeed Asadi Bagloee et al. "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies". In: *Journal of modern transportation* 24.4 (2016), pp. 284–303.

[2] Adnan Shaout, Dominic Colella, and SS Awad. "Advanced driver assistance systems-past, present and future". In: *2011 Seventh International Computer Engineering Conference (ICENCO'2011)*. IEEE. 2011, pp. 72–82.

[3] Ziran Wang et al. "A Survey on Cooperative Longitudinal Motion Control of Multiple Connected and Automated Vehicles". In: *IEEE Intelligent Transportation Systems Magazine* 12.1 (2020), pp. 4–24.

[4] Orit Taubman-Ben-Ari, Mario Mikulincer, and Omri Gillath. "The multidimensional driving style inventory—scale construct and validation". In: *Accident Analysis & Prevention* 36.3 (2004), pp. 323–332.

[5] Dewei Yi et al. "A machine learning based personalized system for driving state recognition". In: *Transportation Research Part C: Emerging Technologies* 105 (2019), pp. 241–261.

[6] Bingzhao Gao et al. "Personalized adaptive cruise control based on online driving style recognition technology and model predictive control". In: *IEEE transactions on vehicular technology* 69.11 (2020), pp. 12482–12496.

[7] Bing Zhu et al. "Personalized Lane-Change Assistance System with Driver Behavior Identification". In: *IEEE Transactions on Vehicular Technology* 67.11 (2018), pp. 10293–10306.

[8] Andrew Phillip Bolduc, Longxiang Guo, and Yunyi Jia. "Multimodel approach to personalized autonomous adaptive cruise control". In: *IEEE Transactions on Intelligent Vehicles* 4.2 (2019), pp. 321–330.

[9] David Silver, J Andrew Bagnell, and Anthony Stentz. "Learning autonomous driving styles and maneuvers from expert demonstration". In: *Experimental Robotics*. Springer. 2013, pp. 371–386.

[10] S. Rosbach et al. "Driving with Style: Inverse Reinforcement Learning in General-Purpose Planning for Automated Driving". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 2658–2665.

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

[12] Liting Sun, Wei Zhan, and Masayoshi Tomizuka. "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 2111–2117.

[13] Xishun Liao et al. "Online Prediction of Lane Change with a Hierarchical Learning-Based Approach". In: *Proceedings 2022 IEEE International Conference on Robotics and Automation*. IEEE. 2022.

[14] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. "Learning driving styles for autonomous vehicles from demonstration". In: *Proceedings - IEEE International Conference on Robotics and Automation* 2015-June.June (2015), pp. 2641–2646.

[15] Maximilian Naumann et al. "Analyzing the Suitability of Cost Functions for Explaining and Imitating Human Driving Behavior based on Inverse Reinforcement Learning". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 5481–5487.

[16] Saurabh Arora and Prashant Doshi. "A survey of inverse reinforcement learning: Challenges, methods and progress". In: *Artificial Intelligence* (2021), p. 103500.

[17] Eiji Uchibe. "Model-free deep inverse reinforcement learning by logistic regression". In: *Neural Processing Letters* 47.3 (2018), pp. 891–905.

[18] Monica C Vroman. "Maximum likelihood inverse reinforcement learning". PhD thesis. Rutgers University-Graduate School-New Brunswick, 2014.

[19] Chao Lu et al. "Transfer Learning for Driver Model Adaptation in Lane-Changing Scenarios Using Manifold Alignment". In: *IEEE Transactions on Intelligent Transportation Systems* (2019).

[20] Zirui Li et al. "Transferable Driver Behavior Learning via Distribution Adaption in the Lane Change Scenario". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 193–200.

[21] Martina Hasenjäger, Martin Heckmann, and Heiko Wersing. "A Survey of Personalization for Advanced Driver Assistance Systems". In: *IEEE Transactions on Intelligent Vehicles* 5.2 (2019), pp. 335–344.

[22] Ziran Wang et al. "Driver Behavior Modeling using Game Engine and Real Vehicle: A Learning-Based Approach". In: *IEEE Transactions on Intelligent Vehicles* 5.4 (2020), pp. 738–749.

[23] Bingzhao Gao et al. "Personalized Adaptive Cruise Control Based on Online Driving Style Recognition Technology and Model Predictive Control". In: *IEEE Transactions on Vehicular Technology* (2020).

[24] Bing Zhu et al. "Typical-driving-style-oriented Personalized Adaptive Cruise Control design based on human driving data". In: *Transportation research part C: emerging technologies* 100 (2019), pp. 274–288.

[25] Wenshuo Wang et al. "A learning-based approach for lane departure warning systems with a personalized driver model". In: *IEEE Transactions on Vehicular Technology* 67.10 (2018), pp. 9145–9157.

[26] Xin Chen et al. "A learning model for personalized adaptive cruise control". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 379–384.

[27] Yanbing Wang et al. "Personalized Adaptive Cruise Control via Gaussian Process Regression". In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. 2021, pp. 1496–1502.

[28] Andrew Y Ng, Stuart J Russell, et al. "Algorithms for inverse reinforcement learning." In: *ICML*. Vol. 1. 2000, p. 2.

[29] Sergey Levine and Vladlen Koltun. "Continuous Inverse Optimal Control with Locally Optimal Examples". In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. ICML'12. 2012, pp. 475–482.

[30] Vinamra Jain, Prashant Doshi, and Bikramjit Banerjee. "Model-free IRL using maximum likelihood estimation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 3951–3958.

[31] Hongbo Gao et al. "Car-following method based on inverse reinforcement learning for autonomous vehicle decision-making". In: *International Journal of Advanced Robotic Systems* 15.6 (2018), p. 1729881418817162.

[32] Zhouqiao Zhao et al. "Personalized Car Following for Autonomous Driving with Inverse Reinforcement Learning". In: *Proceedings 2022 IEEE International Conference on Robotics and Automation*. 2022.

[33] Wenshuo Wang, Junqiang Xi, and J. Karl Hedrick. "A Learning-Based Personalized Driver Model Using Bounded Generalized Gaussian Mixture Models". In: *IEEE Transactions on Vehicular Technology* 68.12 (2019), pp. 11679–11690.

[34] Kyriacos Shiarlis, Joao Messias, and SA Whiteson. "Inverse reinforcement learning from failure". In: *International Foundation for Autonomous Agents and Multiagent Systems*. 2016, pp. 1060–1068.

[35] Deepak Ramachandran and Eyal Amir. "Bayesian Inverse Reinforcement Learning." In: *IJCAI*. Vol. 7. 2007, pp. 2586–2591.

[36] Brian D Ziebart et al. "Maximum entropy inverse reinforcement learning." In: *AAAI*. Vol. 8. 2008, pp. 1433–1438.

[37] Fan Liu and Yong Deng. "Determine the number of unknown targets in Open World based on Elbow method". In: *IEEE Transactions on Fuzzy Systems* (2020).

[38] Jongmin Lee et al. "Monte-Carlo Tree Search for Constrained POMDPs." In: *NeurIPS*. 2018, pp. 7934–7943.