Smooth and Stopping Interval Aware Driving Behavior Prediction at Un-signalized Intersection with Inverse Reinforcement Learning on Sequential MDPs

Shaoyu Yang[1], Hiroshi Yoshitake[2], Motoki Shino[2], and Masamichi Shimosaka[1]

***Abstract*— Driving behavior modeling (DBM) is widely used in the intelligent vehicle field to prevent accidents, which predicts actions that vehicles should take to optimize safe driving behaviors. According to some statistics, accidents easily happen at un-signalized intersections. Modeling driving behavior at such places is of great importance. However, current inverse reinforcement learning-based DBM methods fail to predict proper behaviors at the un-signalized intersections in the aspects of smoothness and stopping behavior by just using a single Markov decision process (MDP). We propose a novel sequential MDPs approach to model the driving behavior at the un-signalized intersections to solve the problems. Our approach decomposes the target behavior through the un-signalized intersections into three parts and models each decomposition's driving behaviors with appropriate time durations by a stopping-time-interval distribution through dynamic programming. Experiments on real driving data show that the proposed method achieved a better result and successfully improved the smoothness and stopping awareness of the planned driving path compared to the baselines.**

I. Introduction

Intelligent vehicles have been attracting attention in recent years for various driving tasks and safety guarantees. On the one hand, accidents are more likely to happen at intersections on residential roads, especially the intersections that do not have any signal. Driving behavior modeling (DBM) is one of the techniques to ensure the safety of proper driving behavior when passing those un-signalized intersections to prevent pervasive driving and accidents, which can be used in applications such as the advanced driver assistance system (ADAS) and the driver support system to avoid potential dangers [1]. On the other hand, driving through the intersection is a very complicated task. Since vehicles come from various locations and perform different behaviors (e.g., going straight, turning left, or right), drivers have to be aware of other cars and pedestrians sometimes. Therefore, the prevention of such accidents is known to be a significant problem in recent years.

Accidents are likely to occur if vehicles enter the intersection rashly due to the low visibility at the un-signalized intersections. To avoid such accidents, we found that the desired driving behaviors for passing the un-signalized intersections will be "stop", "watch", and "pass". Consequently, decreasing the speed before entering the intersections and stopping at the temporary stop line to check the surrounding environment is requested by most countries and regions. In this research, we aim to model this driving behavior.

Electronic pedestrian protection (EPP) [17] is a system to ensure pedestrians' safety. However, this system works only when the accident has already happened. It cannot prevent accidents from happening efficiently due to the drawbacks of the sensor's accuracy and the short braking distances. By analyzing the causes of accidents at the intersections, we found that those drivers who encounter the accidents often drive vehicles through specific locations at improper speeds. As a result, it is necessary to build a robust model towards a safe driving behavior with reasonable velocities at particular positions by taking the accelerations and decelerations into consideration.

As a prominent approach for this issue, inverse reinforcement learning (IRL) is gaining popularity in recent DBM tasks [14] [12]. Figure 1 demonstrates the basic framework of how IRL is applied to DBM at the un-signalized intersections The reward functions of a Markov Decision Process (MDP) will be obtained from real driving data by IRL, which returns a high reward for the desired driving behaviors at specific locations of the target intersection. The trained MDP will then be used for planning the expected driving behaviors according to the obtained reward functions.

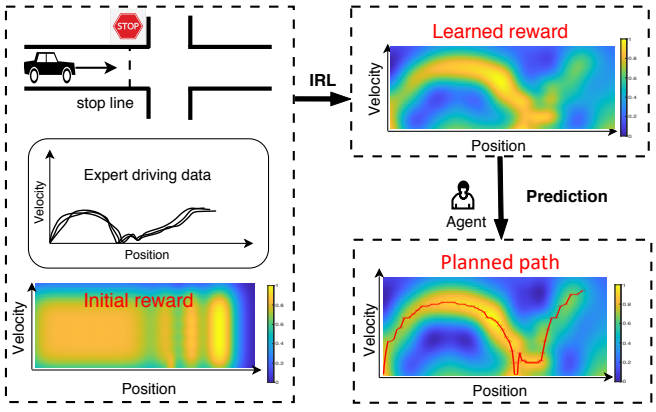


Fig. 1: IRL based DBM at the un-signalized intersection

However, path planning has some problems at such un-signalized intersections by current single-MDP IRL methods. When planning behavior by maximizing the total reward, the

[1]Authors are with the Department of Computer Science, Tokyo Insititute of Technology, Tokyo, Japan. E-mail:{yang, simosaka}@miubiq.cs.titech.ac.jp

[2]Authors are with the Department of Human and Engineered Environmental Studies, The University of Tokyo, Chiba, Japan. E-mail: hyoshitake@edu.k.u-tokyo.ac.jp and motoki@k.u-tokyo.ac.jp

car tends to stop for a long time around the stop line to earn as much reward as possible. While by stochastic policy from reward expectations, stopping time around the stop line becomes shorter but unusual accelerations will happen. In other words, it is difficult for methods based on the single MDP to maintain a balance between the stability of the speed and rationality of the stopping behavior around the stop line. Both cases are abnormal with human driving behavior [16] [18].

To pursue the decent stopping behavior and good smoothness property of the DBM at the un-signalized intersections, we propose a novel sequential MDPs approach. To improve the stopping behavior, we divided the target intersection according to the stop line's position. A sequential MDPs structure with the stopping-time-interval distribution is applied to the intersection partitions for planning the path independently and separately. To make the driving path smoother, we also considered using the Viterbi path's property and a more profound feature functions design, which considers the speed difference of a pair of states. In this sense, it is possible to generate a driving path close to human performances smoothly.

The contributions of this paper are as follows:

- Applying a stopping-time-interval distribution, which better simulates the stopping behavior, increases the awareness of the appropriate stopping behavior around the stop line for the path planning process at the un-signalized intersections.
- The sequential MDPs structure enables the agent to plan a smoother and closer driving path to expert drivers at the un-signalized intersections.

Related work:

Driving pattern matching: For planning the expected expert driving path, a regression model [11] was proposed to model the acceleration behaviors of drivers. But it is very likely to be overfitted with only a limited number of training data. Some approaches [6] [15] [3] used Hidden Markov Model (HMM) to estimate behaviors through vehicle dynamics. However, it is hard to deploy it into diverse environments. Another approach [19] models driving behavior at the intersections, and some possible driving path expectations are generated from topology. Then Dynamic Bayesian Networks (DBN) is used as an estimator to select the path. Although this approach can be applied to some specific environmental contexts, the hierarchical structure is time-consuming and sometimes cannot keep the driving path's anticipated smoothness. To smoothen the driving path, Han et al. [7], proposed a Bézier curve-based path planner. But as for some specific behaviors like stopping behavior at the intersections, this method still couldn't make a satisfying reaction towards certain behaviors.

Imitation learning: Imitation learning [8] is an approach to learn the policy of state-action pairs in a supervised manner from human demonstrations. Behavior cloning [10] is one of the methods in this category. However, Behavior cloning methods suffer from inaccuracies when the environment changes, and those inaccuracies have influences on the safety of the driving. Methods based on computer vision [13] [4] [2] [9] take driving videos as training demonstrations to learn policy directly. But, videos information is hard to process and difficult to sense the driving course's geometrical information. Fine-Grained driving behavior modeling approaches discretize the state space for driving course [14] [12], then use inverse reinforcement learning to learn the reward function with given environmental features for the Markov decision process (MDP) from expert driving data. The driving path is then generated by considering the total earned reward, which effectively solves the inaccuracies of the unseen environment. But it fails to consider certain behaviors when planning over a broader spectrum of driving actions. Our previous work [18] briefly introduces the concept of application of the sequential MDPs in the situation of the un-signalized intersections. However, it lacks a thorough experiment and a more confident result analysis.

The rest of the paper is organized as follows: Section 2 describes the baselines of a single MDP-based DBM. Section 3 describes our work of sequential MDPs at the un-signalized intersections. Section 4 presents the experimental results in terms of smoothness and stopping behavior of the planned driving path. Finally, section 5 provides concluding remarks and future work.

II. Problem settings and single-MDP IRL

This section describes the problem settings of our research and its baselines, single-MDP IRL, then also shows the critical issues raised by the baselines.

A. Problem settings

In this research, we want to model safe driving behavior specifically at the un-signalized intersections. And we realize the importance of velocity at each position when crossing the intersections. Thus the suitable acceleration driving actions are crucial to avoid pervasive driving. Thanks to the advanced navigation system today, it is easy to obtain the road geometry information before entering the un-signalized intersection. Therefore we do not have to plan a long-distance and long-time driving path but focus on the target intersection's length. Based on the assumptions mentioned above, we aim to model the proper velocity for each position concerning the time when passing the target intersection, as shown in Figure 2.

The dynamics of the vehicle is defined for designing the Markov decision process. $\boldsymbol{x}_t \in \mathbb{R}^2$ is a continuous state at the time t represented by a pair of position, p_t, and velocity, v_t, as $\boldsymbol{x}_t = (p_t, v_t)^\top$. Action, $u_t \in \mathbb{R}$, describes acceleration and deceleration behaviors at time t. The assumed dynamics of transitions among states is expressed as

$$\boldsymbol{x}_{t+\Delta t} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \boldsymbol{x}_t + \begin{pmatrix} 0 \\ \Delta t \end{pmatrix} u_t. \tag{1}$$

Then we discretize the continuous state space, x_t, and action space, u_t, into discrete state space, S, and discrete action space, A, in a certain way we will discuss in detail in the experiment section. A MDP, $< S, A, T, R >$, can be applied

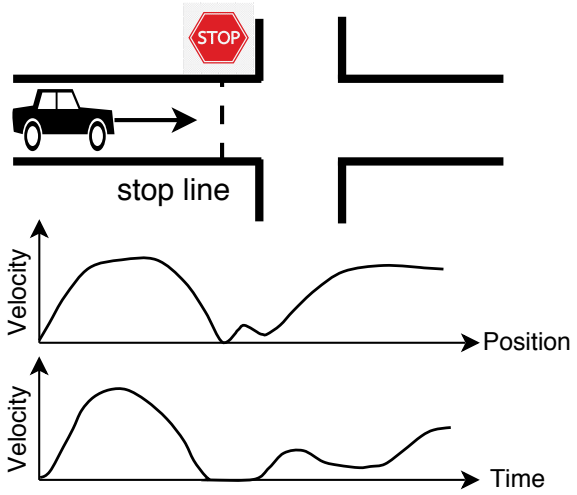


Fig. 2: Modeling target

to represent the problem. Transition, T, is a probability distribution among states obtained from the vehicle dynamics, $P(s' \mid s, a) \mapsto \{0, 1\}$, where $s \in S$ represents the current state and $s' \in S$ is the next state by taking action $a \in A$; and R denotes the reward function. For each state $s \in S$, we obtain its reward by a linear function defined as follows,

$$R(s) = \boldsymbol{w}^\top \boldsymbol{f}(s), \tag{2}$$

where $R(s) \in \mathbb{R}$, and $\boldsymbol{w} \in \mathbb{R}^n$ denotes the weights for feature functions, $\boldsymbol{f}(s) \in \mathbb{R}^n$, designed manually.

Regarding the design of feature functions, we mainly consider how to encode the intersection's geometrical information. Therefore, as one of the simple but effective approaches to designing rewards, we employ Gaussian kernels. These are placed in specific positions to emphasize the importance of acceleration and deceleration behaviors. We will discuss more details in the experiment part.

B. Standard single-MDP

To optimize the reward function, Maximum entropy IRL [20] is utilized to train weights, $\boldsymbol{w}$. This approach's core concept is to maximize the likelihood of given expert samples, what we call paths, in the rest of the paper. Firstly, the likelihood of a path, $p(\zeta^{(i)} \mid \boldsymbol{w})$, is defined as

$$p(\zeta^{(i)} \mid \boldsymbol{w}) = \frac{\exp\left(\sum_{t=1}^{T^{(i)}} R(s_t^{(i)})\right)}{Z^{(i)}(\boldsymbol{w})}, \tag{3}$$

where $\zeta^{(i)} = \{s_1^{(i)}, a_1^{(i)}, s_2^{(i)}, a_2^{(i)} \ldots, s_T^{(i)}\}$ denotes a path from driving dataset, D, which contains N paths performed by expert drivers, whose time duration, $T^{(i)}$, is expressed by $|\zeta^{(i)}| = T^{(i)}$. $Z^{(i)}(\boldsymbol{w})$ means the partition function. The optimal weights, $\boldsymbol{w}^*$, is obtained by minimizing the negative log-likelihood of expert data with a regularization term, $\Omega(\boldsymbol{w})$,

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}}(-\sum_{i=1}^{N} \log p(\zeta^{(i)} \mid \boldsymbol{w}) + \Omega(\boldsymbol{w})). \tag{4}$$

Planning driving paths from the learned model is essential for driving behavior modeling. In this paper, a specific time duration is used to restrict the total time for performing path planning, called finite horizon path planning. There are two basic kinds of paths to be planned.

1) Viterbi path: The Viterbi path, ζ^*, is an estimate of the series of states from MDP, which has a maximum accumulated reward or likelihood within a finite time duration.

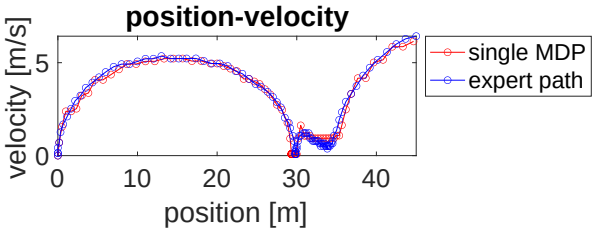
$$\begin{aligned} \zeta^* &= \underset{\zeta \in \Xi}{\operatorname{argmax}} R(\zeta \mid \boldsymbol{w}) \\ &= \underset{\zeta \in \Xi}{\operatorname{argmax}} \sum_{s \in \zeta} R(s \mid \boldsymbol{w}), \end{aligned} \tag{5}$$

where a set of trajectories, Ξ, contains instances whose duration are all equal to t, $\Xi = \{\zeta \mid |\zeta| = t\}$.

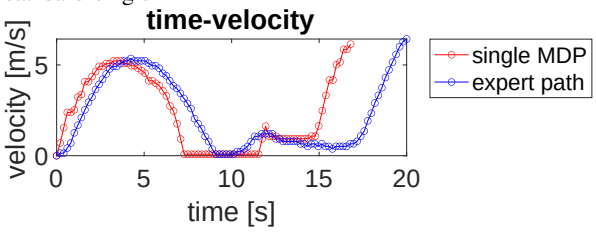
2) Stochastic path: Since there are a number of possible paths for the planning, and each of them has its probability, the stochastic path will be generated from the reward expectations in a probabilistic way.

C. Existing problems of standard single-MDP IRL

Comparing the generated Viterbi path by the standard single-MDP IRL with the expert driving data, we found that the stopping time is extremely longer than experts' performances, as shown in Figure 3. This is because the reward at the temporary stop line is higher than in other positions. Consequently, the agent tends to earn more rewards by stopping at the temporary stop line for a long time.



(a) position-velocity graph of generated Viterbi path by standard single-MDP IRL

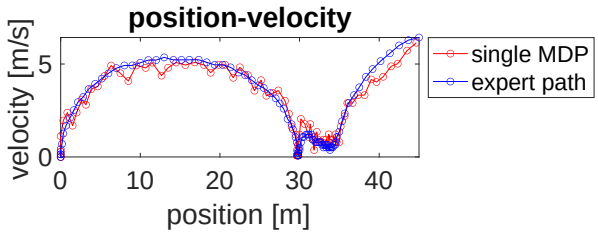


(b) time-velocity graph of generated Viterbi path by standard single-MDP IRL

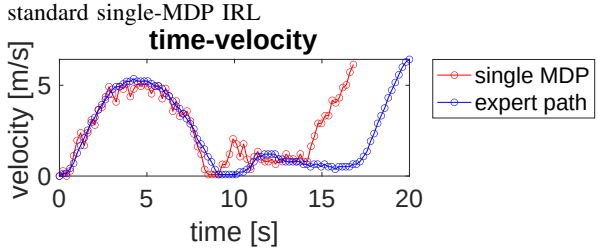
Fig. 3: Long stopping time behavior generation by standard single-MDP IRL

In order to solve the problem brought by the Viterbi path, some studies have adopted the stochastic path instead of the Viterbi path when doing the path planning [16]. However, due to the stochastic property, unusual accelerations will happen, as illustrated in Figure 4. That will affect the smoothness of driving behaviors significantly. Plus, for some

generated stochastic paths, stopping behavior will be ignored and that is very dangerous.



(a) position-velocity graph of generated stochastic path by standard single-MDP IRL



(b) time-velocity graph of generated stochastic path by standard single-MDP IRL

Fig. 4: Unusual accelerations driving behavior generation by standard single-MDP IRL

III. Proposed Method: IRL on Sequential MDPs with Time-Interval Model

A. Overview

This section aims to improve the path planning performance at the un-signalized intersection for awareness of proper stopping behavior and trajectory smoothness. To enhance the path planning with satisfying smoothness, we determine to divide the intersection into few steps and choose a reasonable path generation strategy. To keep a good awareness of stopping behavior at the un-signalized intersection, we propose a time-interval model to increase the awareness of the stopping behavior.

B. Intersection decomposition

By analyzing experts' stopping behavior, we notice that they often stop around the temporary stop line before entering the intersection. Hence we consider the idea of stopping range, which is an area around the temporary stop line for vehicles to conduct the stopping behavior. Then we decompose the intersection into three parts: *before*, *during*, and *after* the stopping range. Each part of the decomposition is modeled by an independent MDP defined as follows:

$$\mathrm{MDP}_x =< S_x, A, T_x, R_x >, \tag{6}$$

where x is b, s and a, the MDP_x represent the MDP *before*, *during* and *after* the stopping range, respectively.

C. Probabilistic modeling with sequential MDPs and time-interval distribution.

The second section mentioned the problems caused by the standard single MDP. We will introduce how the proposed method based on the sequential MDP solves the above issues in two aspects.

1) Enhancement of stopping behavior awareness: For simulating the time duration within the stopping range, t_s, precisely, we use a Poisson distribution to model the time duration by the following definition,

$$t_s \sim \mathrm{Poisson}(\lambda), \tag{7}$$

where λ is the parameter for the Poisson distribution.

A typical problem is how to decide the total duration and durations for each part of the decomposition. Time durations are given by equation, $t_{\mathrm{all}} = t_b + t_s + t_a$, where t_{all} indicates the whole duration; t_b, t_s, and t_a are the time durations *before*, *during* and *after* the stopping range, respectively. t_{all} should maximize the likelihood of the simulated driving path, $\hat{\zeta}$,

$$\begin{aligned} t_{\mathrm{all}} &= \operatorname*{argmax}_{t_{\mathrm{all}}} p(\hat{\zeta}|\mathrm{MDP_b}, \mathrm{Poisson}, \mathrm{MDP_a}) \\ &= \operatorname*{argmax}_{t_b,t_s,t_a} p(\hat{\zeta_b}|\mathrm{MDP_b}, |\hat{\zeta_b}| = t_b) p(t_s|\mathrm{Poisson}) \\ &\quad p(\hat{\zeta_a}|\mathrm{MDP_a}, |\hat{\zeta_a}| = t_a). \end{aligned} \tag{8}$$

where $\hat{\zeta_b}$ and $\hat{\zeta_a}$ are paths generated by MDP_b and MDP_a. And t_s is the time duration of the path $\hat{\zeta_s}$ generated by MDP_s. This problem can be solved by dynamic programming efficiently in algorithm 1.

Algorithm 1 Calculating time partition by dynamic programming

Input: max trajectory length maxT, min trajectory length before stopping range $\mathrm{minT_b}$, min trajectory length after stopping range $\mathrm{minT_a}$, trained MDP and Poisson models $\mathrm{MDP_b}$, $\mathrm{MDP_a}$, Poisson
Output: time partitions $\mathrm{t_b}$, $\mathrm{t_s}$, $\mathrm{t_a}$
Function: FindTimePartitions(maxT, $\mathrm{minT_b}$, $\mathrm{minT_a}$, $\mathrm{MDP_b}$, $\mathrm{MDP_a}$, Poisson)
// Caching the posterior for each partition for all possible time durations
for $t \leftarrow \mathrm{minT_b}$; $t \leq \mathrm{maxT} - \mathrm{minT_a}$; $t \leftarrow t + 1$ **do**
  $f_{\mathrm{b}}(t) \leftarrow p(\hat{\zeta_b}|\mathrm{MDP_b}, |\hat{\zeta_b}| = t)$
end for
for $t \leftarrow 0$; $t \leq \mathrm{maxT} - \mathrm{minT_a} - \mathrm{minT_b}$; $t \leftarrow t + 1$ **do**
  $f_{\mathrm{s}}(t) \leftarrow p(\hat{\zeta_s}|\mathrm{Poisson}, |\hat{\zeta_s}| = t)$
end for
for $t \leftarrow \mathrm{minT_a}$; $t \leq \mathrm{maxT} - \mathrm{minT_b}$; $t \leftarrow t + 1$ **do**
  $f_{\mathrm{a}}(t) \leftarrow p(\hat{\zeta_a}|\mathrm{MDP_a}, |\hat{\zeta_a}| = t)$
end for
// Calculating time partitions by dynamic programming
for $t \leftarrow \mathrm{minT_b}$; $t \leq \mathrm{maxT} - \mathrm{minT_a}$; $t \leftarrow t + 1$ **do**
  $f_{\mathrm{bs}}(t) \leftarrow \max_{\mathrm{t_b},\mathrm{t_s}} f_{\mathrm{b}}(\mathrm{t_b}) f_{\mathrm{s}}(\mathrm{t_s})$, $t = \mathrm{t_b} + \mathrm{t_s}$
end for
for $t \leftarrow \mathrm{minT_b} + \mathrm{minT_a}$; $t \leq \mathrm{maxT}$; $t \leftarrow t + 1$ **do**
  $f_{\mathrm{bsa}}(t) \leftarrow \max_{\mathrm{t_{bs}},\mathrm{t_a}} f_{\mathrm{bs}}(\mathrm{t_{bs}}) f_{\mathrm{a}}(\mathrm{t_a})$, $t = \mathrm{t_{bs}} + \mathrm{t_a}$
end for
$\mathrm{t_a}, \mathrm{t_{bs}} \leftarrow \operatorname{argmax}_{\mathrm{t_a},\mathrm{t_{bs}}} f_{\mathrm{bsa}}(t), t = \mathrm{t_a} + \mathrm{t_{bs}}$
$\mathrm{t_b}, \mathrm{t_s} \leftarrow \operatorname{argmax}_{\mathrm{t_b},\mathrm{t_s}} f_{\mathrm{bs}}(\mathrm{t_{bs}}), \mathrm{t_{bs}} = \mathrm{t_b} + \mathrm{t_s}$
return $\mathrm{t_b}$, $\mathrm{t_s}$, $\mathrm{t_a}$

2) Enhancement of driving path smoothness: In the single-MDP method, both Viterbi and stochastic algorithms are used for path planning for the driving behavior modeling,

but it is hard to keep the smoothness while also retain the good stopping behavior. In the proposed method, we only use the Viterbi path for planning instead of using the stochastic path. The first reason is that the Viterbi path is unique if the time horizon is fixed, which eliminated the uncertainty. On top of that, due to the property of stochastic path, unusual accelerations and decelerations always happen, which highly affects the comfort of driving. Lastly, we usually generate hundreds of stochastic paths and pick up some or take the mean values of them. However, this process is time-consuming, and the qualities of the stochastic paths are too heterogeneous.

D. Inference process for planning the driving behavior

The only problem left to be solved is that the way to connect three Viterbi paths obtained from the trained sequential MDPs. The velocities at the connection boundaries have to be considered, as shown in Figure 5. Currently, we use one intuitive way by taking the average speed of each boundary position from the expert demonstrations. Finally, the full driving path will be planned by combining three generated Viterbi paths together.

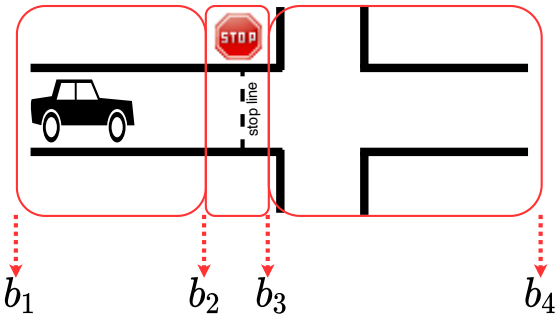


Fig. 5: There are four velocities need to be determined for one intersection noted by b_1, b_2, b_3, and b_4, respectively. Each velocity is calculated by the average speed when expert drivers tend to perform at this position.

E. Model optimization

A training dataset, D, contains n expert trajectories, $D = \{\zeta^{(1)}, \zeta^{(2)}, \dots \zeta^{(n)}\}$, is divided into three sub-datasets according to the intersection decomposition as the sub-dataset before the stopping range, D_b, the sub-dataset during the stopping range, D_s, and the sub-dataset after the stopping range, D_a. For example, the sub-dataset D_s can be expressed by $D_s = \{\zeta_s^{(1)}, \zeta_s^{(2)}, \dots \zeta_s^{(n)}\}$, where $\zeta_s^{(i)}$ is a series of states from the trajectory, $\zeta^{(i)}$, in case those states are inside the stopping range. We train each MDP from the sequential MDPs model by the maximum entropy IRL [20] individually.

Besides, the parameter for the Poisson distribution, λ, is estimated by the sub-dataset D_s alone,

$$\hat{\lambda} = \frac{\sum_{i=1}^{n} |\zeta_s^{(i)}|}{n}. \tag{9}$$

IV. Experiment

A. Experimental purpose

The experiment was carried out at an un-signalized intersection. We collected expert driving data and trained the driving behavior model based on those data. We wanted to see whether the stopping behavior and smoothness of planned paths are improved by the proposed method or not.

B. Dataset

A Honda Civic is used as the experimental vehicle, and we equip the car with a driving recorder (Finefit Design, Tough More-Eye) to record the front views and web cameras (Logitech, C930e) to film drivers' status. A GNSS antenna (Hemisphere, AtlasLink) is used to record the vehicle's positions to calculate the velocity and accelerations. The experiment was conducted in an un-signalized intersection with six different scenes from A to F as the Table I shows. cc (L) and cc (R) mean left corner cut and right corner cut, respectively.

TABLE I: Six scenes of collected data

Scene name	Road width [m]	cc (L)	cc (R)
A	6.0	Blind	Blind
B	3.5	Blind	Blind
C	3.5	Available	Blind
D	3.5	Available	Available
E	3.5	Blind	Available
F	6.0	Available	Available

Figure 6 illustrates an example of an experimental environment when the vehicle crosses through the intersection with scene C. We collected driving data from two driving

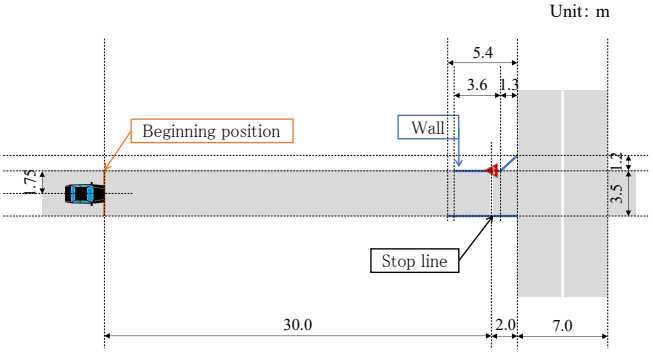


Fig. 6: The experiment settings for scene C, where road width is 3.5 meter with the left corner cut and without the right corner cut.

instructors, who drove 15 times in each scene. The data of one of the instructors were used as training and testing data for this experiment.

C. Fine-grained state space discretization

The driving data are sampled at 5 [Hz] for the timestamp. We also discretize state space and action space for the training purpose. The position is discretized from 0 [m] to 50 [m] at 0.16 [m] intervals into 300 parts, and speed from 0 [m/s] to 7 [m/s] at 0.2 [m/s] intervals into 35 parts.

Therefore, the total number of states is $10,500$. Moreover, we control the acceleration from -4.0 $[\mathrm{m/s^2}]$ to 4.0 $[\mathrm{m/s^2}]$ to make the states' transitions from vehicle dynamics.

D. Feature design

We design the feature functions, $\boldsymbol{f}(s)$, manually. We consider (1) reducing speed at the beginning and end of each decomposition, (2) avoiding high speed or slow speed for the overall course, (3) zero velocity at the temporary stop line. And those features are represented by two-dimensional Gaussian kernels with their positions in discretized state space. Figure 7 shows the initial reward of feature designed of above statement, which takes the weights of equation (2) as all one, $\mathbf{1}$.

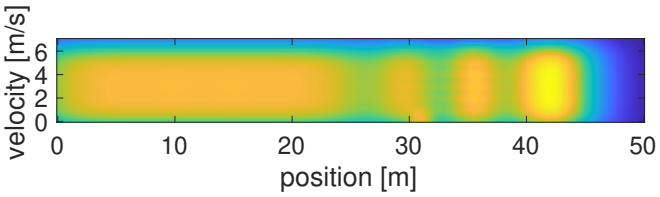


Fig. 7: Initial reward.

To tackle large, improper accelerations at some positions of the un-signalized intersections, we apply a new feature function that takes the difference of velocity of a pair of states into consideration [16]. We call it bi-potential features with the following reward representation,

$$R(s, s') = \boldsymbol{w}^\top \boldsymbol{f}(s, s'). \tag{10}$$

E. Comparison methods

Sequential MDPs: In sequential-MDPs methods, we train the models on the decomposed intersection with standard IRL feature functions and bi-potential feature functions, called standard sequential MDPs (Sequential) and bi-potential sequential MDPs (Sequential-Bi), respectively.

Single MDP: In this category, only one MDP is used to model the whole intersection driving behavior. We also train the models with different types of feature functions. The following paths are used to compare: (1) Single-V, Viterbi path generated from trained standard single MDP. (2) Single-S, stochastic path generated from trained standard single MDP. (3) Single-Bi-V, Viterbi path generated from trained bi-potential single MDP. (4) Single-Bi-S, stochastic path generated from trained bi-potential single MDP.

F. Evaluation metrics

In physics, jerk is the rate at which an object's acceleration changes with respect to time. So it is frequently used as a method to evaluate the smoothness of the driving path. The mean squared jerk, MSJ, for a trajectory, ζ, is

$$\mathrm{MSJ} = \frac{\sum_{t=1}^{(|\zeta|-1)} (\mathrm{acc}_{t+1} - \mathrm{acc}_t)^2}{|\zeta| - 1}, \tag{11}$$

where acc_t represents the acceleration of path, ζ, at time t.

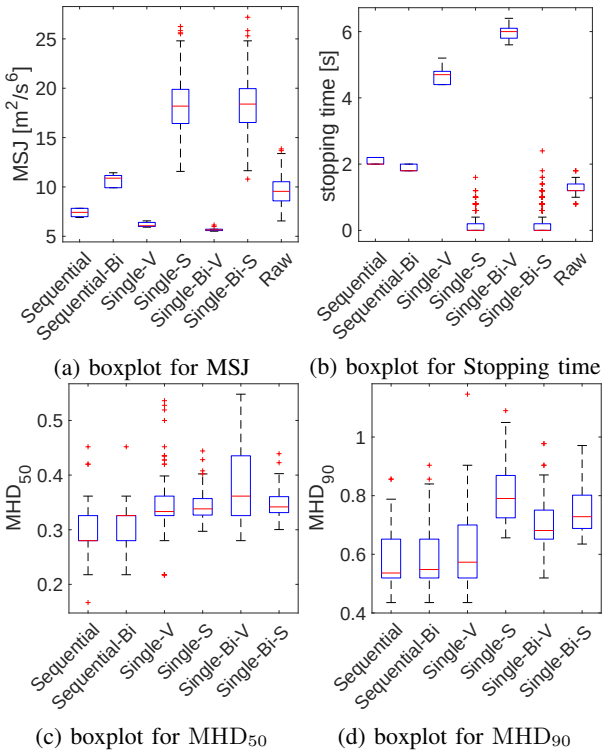
We also evaluated the stopping time near the temporary stop line to understand whether the stopping behavior is appropriate. The stop time should be the time within the stop range, and its speed should be less than 0.2 m/s at any time.

We applied modified Hausdorff distance (MHD) [5] to see the similarity of planned paths with paths performed by expert drivers. MHD matches two sequences with different lengths and evaluates their difference. In this experiment, we evaluated paths in a position-velocity manner and set the MHD parameter α equals to 0.5 or 0.9. α means the α percentile of the distances (e.g., $\alpha = 0.5$, the median of the distances). And we represent them by MHD_{50} and MHD_{90}.

Due to the limited number of data, the test of significance (t-test) is a formal procedure for comparing observed results with the claims, the truth of which is being assessed or not.

G. Experimental results

The experimental results were obtained from 6 fold cross-validation, which leaves one scene out for testing and the rest for training. Figure 8 illustrates experiment results of boxplots for proposed sequential-MDPs methods, single-MDP methods, and raw expert data.



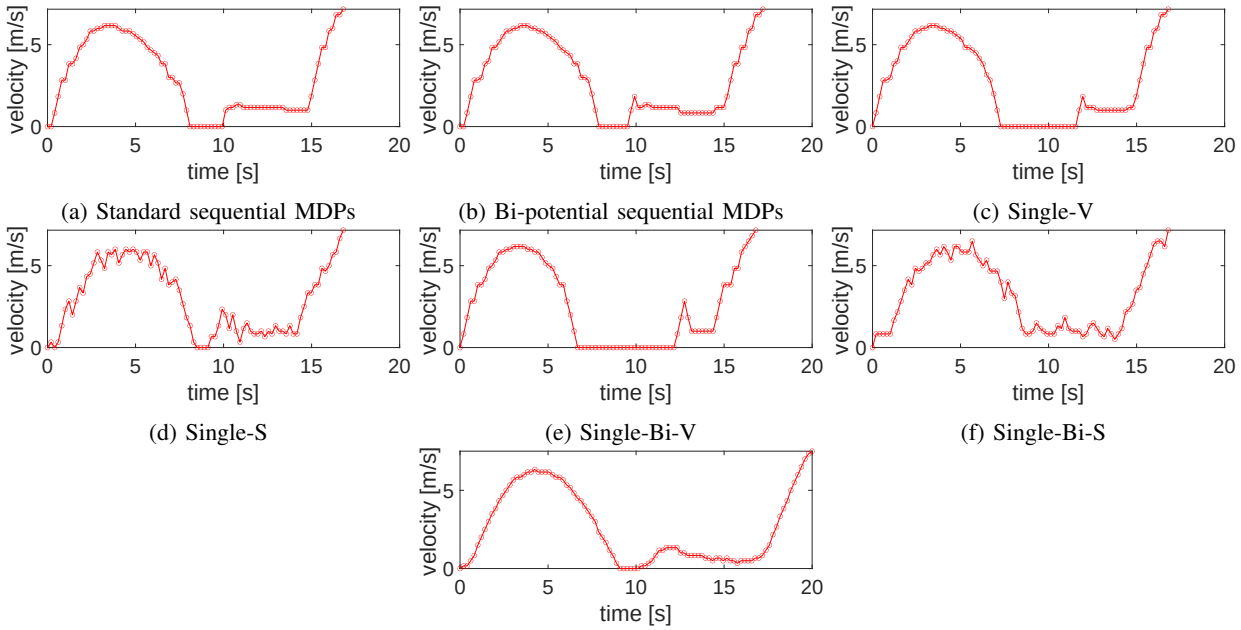
(a) boxplot for MSJ (b) boxplot for Stopping time

(c) boxplot for MHD_{50} (d) boxplot for MHD_{90}

Fig. 8: boxplots for evaluation results

Table II gives the errors of comparison methods with the test data in smoothness and stopping behavior evaluation. Sequential-MDPs methods have achieved the best performances among others.

Table III shows the MHD_{50} and MHD_{90} results, respectively. Sequential MDPs with the bi-potential IRL have achieved the best result, but it is almost the same as the performance of Single-V.



(a) Standard sequential MDPs (b) Bi-potential sequential MDPs (c) Single-V

(d) Single-S (e) Single-Bi-V (f) Single-Bi-S

(g) expert path

Fig. 9: Planned paths and raw expert path

TABLE II: Evaluation errors with test data

Method	MSJ error$[m^2/s^6]$	Stopping time error$[s]$
Sequential	2.30± 1.43	0.80 ± 0.27
Sequential-Bi	**1.62 ± 1.13**	**0.60 ± 0.27**
Single-V	3.47 ± 1.55	3.42 ± 0.31
Single-S	8.59 ± 1.66	1.10 ± 0.23
Single-Bi-V	3.96 ± 1.52	4.68 ± 0.31
Single-Bi-S	8.64 ± 1.60	1.07 ± 0.22

TABLE III: MHD results

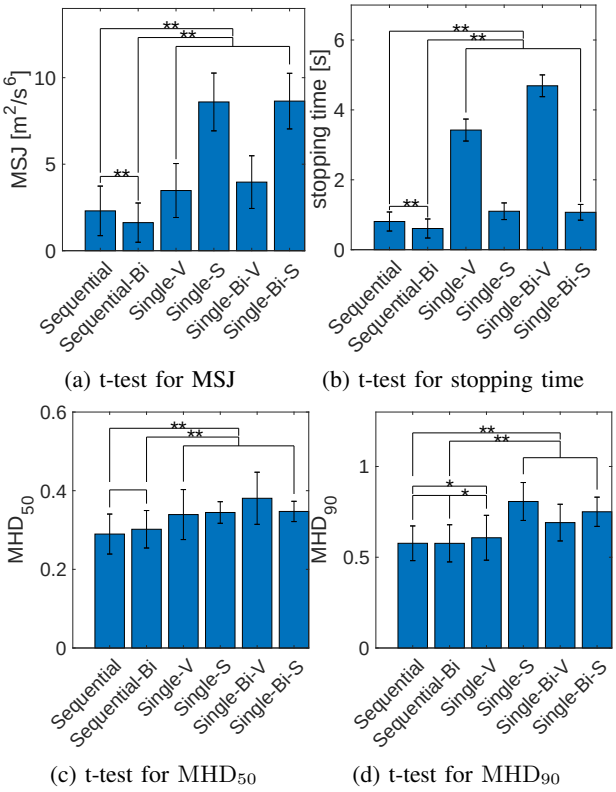
Method	MHD_{50}	MHD_{90}
Sequential	**0.28± 0.05**	0.57 ± 0.09
Sequential-Bi	0.30 ± 0.04	**0.57 ± 0.10**
Single-V	0.33 ± 0.06	0.60 ± 0.12
Single-S	0.34 ± 0.02	0.80 ± 0.10
Single-Bi-V	0.38 ± 0.06	0.69 ± 0.10
Single-Bi-S	0.34 ± 0.02	0.75 ± 0.08

Due to the limited number of testing data, we conducted statistical hypothesis testing (two-sample t-test) to verify the real difference among those methods. First, we set up the null hypothesis, H_0, as the means of two evaluation results are equal to each other. The t-test results shown in Figure 10 represent whether the null hypothesis, H_0, should be rejected or not. Asterisks represent whether p-value, p, is less than a predetermined significance levels. "**" represents $p < 0.001$, and "*" indicates $p < 0.1$.

We also checked the planned driving paths by each comparison method. The smoothness and stopping behavior for the proposed sequential-MDPs methods are greatly improved as shown by Figure 9.

H. Discussion

We want to discuss the experimental results in both quantitative and qualitative ways. In terms of the smoothness of



(a) t-test for MSJ (b) t-test for stopping time

(c) t-test for MHD_{50} (d) t-test for MHD_{90}

Fig. 10: two-sample t-test results: figures show the bar graph of mean values of evaluation results with error bars for each method.

the planned paths, sequential-MDPs methods have achieved the best result, especially for bi-potential sequential MDPs, which improved the error by at least 53%. As for the stopping behavior, sequential MDPs methods also increased the stopping awareness by at least 46%. The most important fact is that sequential-MDPs methods successfully satisfies both sides of decent stopping behavior and good smoothness property, which was not fulfilled by previous works of DBM at the intersection [12] [16] [14]. Regarding the MHD results, sequential-MDPs methods also outperformed all comparison methods that the planned paths were most closer to experts' data.

The t-test results showed that sequential-MDPs approaches outperformed other methods in smoothness and stopping behavior evaluation with statistical significance. Also, it is hard to deny that standard sequential MDPs and bi-potential sequential MDPs have different performances in MHD evaluation regardless of their statistical figures' differences.

Figure 9a and 9b are paths generated by the sequential-MDPs approaches, whose smoothness and stopping behavior are close to the expert data shown in Figure 9g. In contrast, Figure 9c-9f illustrate the paths planned by the single MDP approach. They do not have proper stopping behavior and also do not meet the satisfactory smoothness property. It is noteworthy that Figure 9b has a sudden acceleration and deceleration behavior before entering the intersection. This abnormal behavior might be due to the property of bi-potential reward that constant acceleration tends to be rewarded. The vehicle keeps the acceleration before decreasing the speed.

V. CONCLUSION

We proposed a sequential-MDPs approach to model the driving behavior at the un-signalized intersections, which effectively improved the smoothness and stopping awareness of the planned paths. Moreover, we evaluated our model by running the experiments on real driving data. The proposed sequential MDPs with standard or bi-potential features achieved the best results among other comparison methods. We also conducted the t-test to verify the evaluation results to guarantee confidence in a limited number of data. As for the limitation of this work, the computational cost will increase if we expand the number of segmentation and course distances. The future work will focus on more generic applications of driving behavior modeling to diverse driving tasks and dynamic environments to realize the long-term driving behavior modeling.

REFERENCES

[1] N. AbuAli and H. Abou-zeid, "Driver behavior modeling: Developments and future directions," *International Journal of Vehicular Technology*, 2016.

[2] P. Aditya *et al.*, "Exploring data aggregation in policy learning for vision-based urban autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] N. Akai *et al.*, "Driving behavior modeling based on hidden markov models with driver's eye-gaze measurement and ego-vehicle localization," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[4] M. Bojarski *et al.*, "End to end learning for self-driving cars," 2016.

[5] M. . Dubuisson and A. K. Jain, "A modified hausdorff distance for object matching," in *Proceedings of 12th International Conference on Pattern Recognition (ICPR)*, 1994.

[6] V. Gadepally *et al.*, "A framework for estimating driver decisions near intersections," *IEEE Transactions on Intelligent Transportation Systems*, pp. 637–646, 2014.

[7] L. Han *et al.*, "Bézier curve based path planning for autonomous vehicle in urban environment," in *IEEE Intelligent Vehicles Symposium (IV)*, 2010.

[8] A. Kuefler and A. Zisserman, "Imitating driver behavior with generative adversarial networks," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[9] K. Lee *et al.*, "Approximate inverse reinforcement learning from vision-based imitation learning," 2020.

[10] S. Lefèvre *et al.*, "Comparison of parametric and non-parametric approaches for vehicle speed prediction," in *American Control Conference (ACC)*, 2014.

[11] S. Mondal *et al.*, "Modeling driver acceleration behaviour at signalized intersection under mixed traffic environment," *Journal of the Eastern Asia Society for Transportation Studies*, pp. 1761–1776, 2019.

[12] K. Nishi *et al.*, "Fine-grained driving behavior prediction via context-aware multi-task inverse reinforcement learning," in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[13] D. Pomerleau *et al.*, "Alvinn: an autonomous land vehicle ina neural network." in *In Advances in Neural Information Processing Systems (NIPS)*, 1988.

[14] M. Shimosaka *et al.*, "Modeling risk anticipation and defensive driving on residential roads with inverse reinforcement learning," in *International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[15] T. Streubel and K. H. Hoffmann, "Prediction of driver intended path at intersections," in *IEEE Intelligent Vehicles Symposium Proceedings (IV)*, 2014.

[16] K. Tachibana *et al.*, "Driver modeling at unsignalized intersection with stop lines using inverse reinforcement learning (in japanese)," in *Society of Automotive Engineers of Japan (JSAE) Congress (Spring)*, 2020.

[17] J. Tilp *et al.*, "Pedestrian protection based on combined sensor systems," in *International technical conference on the enhanced safety of vehicles (ESV)*, 2005.

[18] S. Yang *et al.*, "Driving behavior modeling at unsignalized intersection with inverse reinforcement learning on sequential MDPs," in *The Society of Instrument and Control Engineers (SI)*, 2020.

[19] J. Zhang and R. Bernd, "Situation analysis and adaptive risk assessment for intersection safety systems in advanced assisted driving," *Autonome Mobile Systeme (AMS)*, pp. 249–258, 2009.

[20] B. D. Ziebart *et al.*, "Maximum entropy inverse reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2008.