

Full Vehicle Trajectory Planning Model for Urban Traffic Control Based on Imitation Learning

Transportation Research Record
1–13© National Academy of Sciences:
Transportation Research Board 2022
Article reuse guidelines:sagepub.com/journals-permissions

DOI: 10.1177/03611981221077263

journals.sagepub.com/home/trrJun Ying¹ and Yiheng Feng¹ 

Abstract

Connected and automated vehicles (CAVs) extend urban traffic control from temporal to spatiotemporal by enabling the control of CAV trajectories. Most of the existing studies on CAV trajectory planning only consider longitudinal behaviors (i.e., in-lane driving), or assume that the lane changing can be done instantaneously. The resultant CAV trajectories are not realistic and cannot be executed at the vehicle level. The aim of this paper is to propose a full trajectory planning model that considers both in-lane driving and lane changing maneuvers. The trajectory generation problem is modeled as an optimization problem and the cost function considers multiple driving features including safety, efficiency, and comfort. Ten features are selected in the cost function to capture both in-lane driving and lane changing behaviors. One major challenge in generating a trajectory that reflects certain driving policies is to balance the weights of different features in the cost function. To address this challenge, it is proposed to optimize the weights of the cost function by imitation learning. Maximum entropy inverse reinforcement learning is applied to obtain the optimal weight for each feature and then CAV trajectories are generated with the learned weights. Experiments using the Next Generation Simulation (NGSIM) dataset show that the generated trajectory is very close to the original trajectory with regard to the Euclidean distance displacement, with a mean average error of less than 1 m. Meanwhile, the generated trajectories can maintain safety gaps with surrounding vehicles and have comparable fuel consumption.

Keywords

vehicle trajectory planning, imitation learning, maximum entropy inverse reinforcement learning, in-lane driving, lane changing

Traditional urban traffic control is mainly conducted from the infrastructure side, such as by traffic signals, which assign green and red phases to different vehicle movements. Human drivers follow the signal indicators and control their vehicles' maneuvers. With connected and automated vehicle (CAV) technology, not only traffic signals but also the CAV trajectories can be controlled, in either a centralized or a decentralized manner, to further improve the efficiency of the intersection operations. CAV trajectory planning is a critical problem in this new control paradigm and is applied to different scenarios, from mixed traffic conditions with human-driven vehicles (HDVs) (1, 2) to autonomous intersection management (AIM) (3) in a full CAV environment. Full trajectory planning includes two dimensions: longitudinal (e.g., in-lane driving) and lateral (e.g., lane changing). Although lane changing is an essential maneuver in many driving scenarios, most existing studies only plan

longitudinal speed/acceleration without considering lane changing (4, 5) or they model the lane changing in a simplified way, assuming the maneuver is done instantaneously without explicitly modeling the vehicle trajectory during lane changing (6). This simplification may result in unrealistic trajectories which cannot be executed at the vehicle level. In addition, it may also affect the following vehicle in the target lane. Inaccurate prediction of the behavior of a cut-in vehicle (i.e., the lane changing vehicle) may cause safety issues. Therefore, the lateral maneuver of lane changing should be coupled with longitudinal speed in the planning process to generate more realistic vehicle trajectories.

¹Lyles School of Civil Engineering, Purdue University, West Lafayette, IN

Corresponding Author:

Yiheng Feng, feng333@purdue.edu

The applications of imitation learning (7, 8) have been fruitful in recent years. Imitation learning aims to mimic human behaviors in a given task, such as CAV trajectory planning. There are several benefits to planning human-like CAV trajectories. First, human driving behaviors are natural baselines for CAVs. Second, human-like CAV trajectories are less disruptive in the mixed traffic condition with HDVs. For example, some existing studies proposed end-to-end learning models (9, 10) to imitate human driving behaviors. Third, unlike other hierarchical trajectory planning models, which need to decide the lane changing time (or position) on the upper level (11), imitation learning integrates the car-following and lane changing maneuvers into a full trajectory planning model. Finally, compared with other deep learning based models (12) which aim to derive the optimal policy, the main objective of imitation learning is to obtain the cost function, which requires much fewer data (i.e., demonstrations) and less time. With the learned cost function, an optimization problem can be formulated to generate smooth and safe human-like trajectories as well as to predict the trajectories of surrounding vehicles.

In this paper, such a full trajectory planning model based on imitation learning is proposed. This work can be briefly summarized as follows. The vehicle trajectory planning is modeled as an optimization problem with specifically designed features in the cost function representing safety, efficiency, and comfort. Maximum entropy inverse reinforcement learning (IRL) is then applied to obtain the weight vector of the cost function based on the training dataset, which includes both in-lane driving and lane changing scenarios. With the parameters learned from the IRL and a given initial state, optimal trajectories are generated by solving the optimization problem. In the case study, the Next Generation Simulation (NGSIM) Lankershim Boulevard Dataset (13) is applied to evaluate the proposed model. Results show that the generated trajectories are very close to the ground truth trajectories. In addition, the safety performance in relation to critical gaps and fuel consumption of the generated trajectories are investigated.

The main contributions of the paper are listed as follows:

1. A trajectory planning framework is proposed that can generate human-like trajectories, including the vehicle speed, acceleration, and heading angle at each timestep with given origin, destination, and arrival time. The model can be easily integrated with either vehicle level control or intersection traffic control.
2. A cost function is designed for the IRL that integrates both in-lane and lane changing scenarios

with consideration of road geometry, which makes up most of the driving scenarios in a road segment.

3. The real-world NGSIM dataset is applied for evaluation. The proposed model can generate trajectories that are very close to human driving behaviors with sufficient safety gaps and comparable energy consumption.
4. The proposed trajectory generation model can also be used to model car-following and lane changing behaviors in microscopic traffic simulation.

The remainder of the paper is arranged as follows. The next section reviews related literature on CAV trajectory planning. The third section introduces the methodologies including the overall framework, trajectory generation optimization problem, feature selection, and the IRL model. The evaluation results from the NGSIM dataset are presented in the fourth section. The final section concludes the work, relates the proposed work to urban traffic control, and lays out future research directions.

Literature Review

Vehicle trajectory planning has been widely studied in the literature. The problem can be generally divided into two categories: in-lane driving (i.e., car following) and lane changing. In past decades, numerous in-lane driving models for HDVs, CAVs, and mixed traffic flow have been proposed. Interested readers can refer to Mahmassani (14) and Zhou et al. (15) for a comprehensive review. For lane changing models, many earlier methods only model the lane changing decision point (whether and when to change lanes) without considering the detailed changing process (16–18). Recently, some studies have proposed to model the detailed lane changing process for CAV trajectory planning. For example, Zhang et al. (19) proposed a dynamic trajectory planning method, in which vehicle maneuvers are decomposed into lane changing and lane keeping maneuvers, or a combination of both. Vehicle state is estimated given the current coordinates and heading within a short time interval. Yang et al. (11) proposed a dynamic lane changing trajectory planning model. The lane changing point was determined first and a cubic polynomial curve was then applied to generate the lane changing trajectory. The parameters of the curve are a function of vehicle heading angle and the final coordinate of the trajectory. Luo et al. (20) formulated the trajectory planning problem as an optimization problem with safety and comfort constraints. Trajectory length and lane change time were determined by the optimization function. All the above

methods consider lane changing as a separate process and need to determine the lane changing start time first. A few recent studies tried to combine the car-following and lane changing behaviors and formulated integrated optimization problems to determine both longitudinal speed and lane assignment simultaneously (21, 22). However, these studies usually model lane changing as a binary or integer variable by assuming that the lane changing process can be completed in one time step.

Trajectory planning is a challenging task for CAVs, as it involves multiple objectives such as safety, efficiency, and comfort. The abovementioned car-following and lane changing methods can only be applied to simplified roadway conditions (e.g., a straight road without curvature) because detailed vehicle dynamics and road geometries are not considered. To plan more realistic trajectories that can be directly applied at the vehicle control level, IRL has been widely applied in recent years. With a specifically designed reward function, IRL can mimic different driving policies and can handle varying roadway conditions and driving scenarios such as curved roads and lane changing and merging maneuvers. Naumann et al. (23) explored the forms of the cost function for different trajectory planning scenarios such as car-following, merging, and lane changing. Separate cost functions for each scenario were proposed and parameters of the cost function were obtained using maximum entropy IRL. Andersen et al. (24) focused on in-lane driving scenarios. Their proposed model could overtake static obstacles in the urban environment. However, the proposed method could only guide the vehicle to make small lateral changes within the lane. Wang et al. (6) separated the trajectory planning problem into longitudinal and lateral decisions. The longitudinal control calculates the gap of the surrounding vehicle and selects a gap to move to in each time step. The lateral control decides whether to maintain the current lane or move to the target lane. A vehicle is assumed to move to the target gap instantaneously once the decision has been made. Best et al. (25) presented a navigation algorithm for autonomous vehicles. The approach can guarantee vehicle safety while changing lanes. The safety of the lane changing maneuver is guaranteed by penalizing the vehicle if it is in the wrong lane. Kuderer et al. (7) applied maximum entropy IRL to learn different driving styles from demonstrations. The proposed model can be applied to both in-lane driving and lane changing behavior. However, the proposed model is not an end-to-end learning model. The trajectory planning process is divided into small time segments. Vehicle initial state and environmental parameters need to be updated for each planning segment. Sharifzadeh et al. (26) combined projection-based IRL and Deep Q-Networks to generate vehicle trajectories. The proposed method is computationally inefficient.

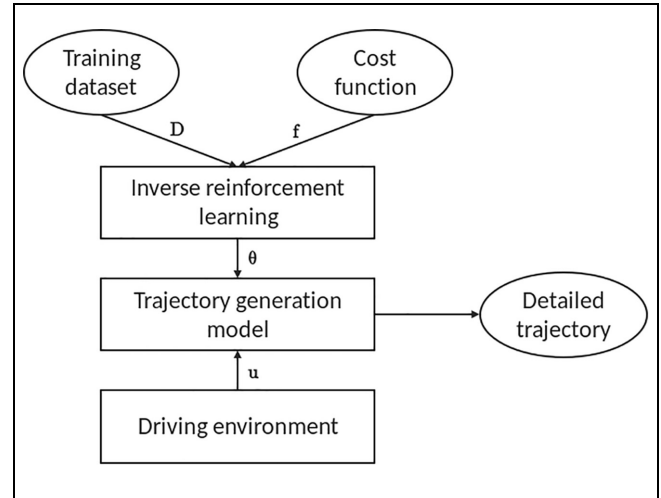


Figure 1. Trajectory planning framework.

Methodology

Problem Statement

The overall flow of the trajectory planning problem is illustrated in Figure 1. First, a cost function (f) is constructed that consists of features representing different aspects of driving behaviors. A training dataset (D) which includes demonstration trajectories is collected and processed. The maximum entropy IRL method is then applied to learn the weight vector θ of the cost function. After obtaining the cost function, the trajectory generation model is formulated as an optimization problem. Given initial state and driving environment parameters (u), final trajectories can be generated by solving the optimization problem. In the following, the trajectory generation optimization problem is first introduced, followed by feature selection in the cost function, then how IRL is used to obtain the optimal weight vector.

The trajectory generation problem is shown in Equation 1. The cost function is presented as $\theta^T f(s, u)$, where θ is the weight vector and s is the set of trajectory points. $s = (s_1, s_2, \dots, s_N)$, where s_i is the trajectory point at timestep i and N is the length of the whole trajectory. Each trajectory point s_i is a vector consisting of five variables: $v_i, a_i, x_i, y_i, \phi_i$, where v_i is the ego vehicle speed at timestep i ; a_i is the acceleration at timestep i ; x_i, y_i is the longitudinal and lateral coordinate respectively at timestep i ; and ϕ_i is the vehicle heading angle at timestep i . u represents the vehicle's initial state and driving environment parameters. The vehicle's initial state includes the initial vehicle speed and acceleration and heading angle, and its initial and terminal position. The driving environment parameters include position of the leading vehicle, speed limit of the road segment, road geometry in relation to lane heading, and position of leading and

following vehicles in the target lane for the lane changing scenario. Feature selection for the objective function and the constraints will be introduced in the following section.

$$\text{minimize}_s \theta^T f(s, u) \quad (1)$$

s.t

vehicle dynamics constraints

Cost Function

The cost function consists of a linear combination of driving features. Different features represent different objectives in trajectory planning. In total, 10 features are included that reflect three main objectives: safety, efficiency, and comfort. The combination of all features including their associated weights defines a driving policy. By minimizing the weighted cost function, the aim is to balance safety, efficiency, and comfort. The proposed features are generic and can be applied to both in-lane driving and lane changing scenarios. For the in-lane driving behavior, the vehicle either follows a leading vehicle or drives in free flow. For the lane changing behavior, the vehicle changes to an adjacent lane and keeps safe distances from surrounding vehicles. The necessity of lane changing in trajectory planning is determined by the initial and terminal positions of the vehicle. If the initial and terminal positions are in the same lane, an in-lane trajectory will be planned. If they are located in different lanes, then a trajectory that consists of the lane changing maneuver will be planned. The terminal position can be obtained from a high-level trajectory planning module (e.g., path planning) of the CAV. It can also come from the infrastructure in a cooperative driving automation environment. For example, the infrastructure provides high-level guidance for CAV as to arrival time and lane assignment, from which the terminal point can be obtained. Note that this study does not consider the circumstances in which the vehicle changes multiple lanes in one trajectory planning period. If such a maneuver is needed, multiple trajectory planning will be conducted. The three selected features in the cost function are elaborated below.

Comfort Cost.

1. Acceleration: $f_1 = \frac{1}{N} \sum_i \frac{|a_i|}{a_{\max}}$, where a_{\max} is the maximum acceleration the vehicle can achieve. f_1 sums up $|a_i|$ of the whole trajectory. By minimizing the total acceleration, uncomfortable motions are penalized, with potential for reduced fuel consumption and emissions.
2. Jerk: $f_2 = \frac{1}{N-1} \sum_i |a_{i+1} - a_i|$. f_2 is introduced to prevent large acceleration changes.

3. Yaw rate: $f_3 = \frac{1}{N-1} \sum_i |\phi_{i+1} - \phi_i|$. The difference between two consecutive heading angles is minimized to keep the curve smooth.

Safety Cost.

1. Safety distance to the leading vehicle in the same lane: $f_4 = |d_{i,des} - d_{i,act}|m_i = |d_{\min} + t_{headway}v_i - d_{i,act}|m_i$. f_4 measures the difference between the desired space headway and the actual space headway. m_i is an indicator variable that measures the existence of the leading vehicle. $m_i = 0$ if there is no leading vehicle and $m_i = 1$ otherwise. d_{\min} is the desired minimum distance, v_i is the vehicle speed at timestep i , $t_{headway}$ is a time constant which represents the driver's reaction time in the car-following model. In this research, $d_{\min} = 7$ m and $t_{headway} = 1.5$ s.
2. Distance to the leading vehicle in the target lane: $f_5 = \frac{1}{N} \sum_i \frac{1}{\alpha + |x_i - x_{i,front}|} |y_i - y_{i,front}|lp_i$. f_5 measures the longitudinal distance between the ego vehicle and the leading vehicle in the target lane during the lane changing process. (x_i, y_i) is the coordinate of the ego vehicle, $(x_{i,front}, y_{i,front})$ is the coordinate of the leading vehicle in the target lane. This feature only influences the trajectory planning during the lane changing process. l is a binary parameter, which equals 1 when the vehicle needs to change its lane during the trajectory planning process, and 0 otherwise. Note that the value of l is determined by the initial and terminal positions of the trajectory planning process. p_i is another binary parameter, which equals 1 if there is a leading vehicle in the target lane at timestep i , and 0 otherwise. After the lane changing is complete, $|y_i - y_{i,front}|$ will be close to zero. Therefore, this feature will not influence the remaining time steps in the trajectory planning.
3. Distance to the following vehicle in the target lane: $f_6 = \frac{1}{N} \sum_i \frac{1}{\alpha + |x_i - x_{i,follow}|} |y_i - y_{i,follow}|lm_i$. f_6 measures the longitudinal distance between the ego vehicle and the following vehicle in the target lane during the lane changing process. Similarly, m_i is a binary parameter, which equals 1 if there is a following vehicle in the target lane at timestep i , and 0 otherwise. f_5 and f_6 guarantee the safety distances to vehicles in the target lane during the lane changing process.
4. Road geometry: $f_7 = \frac{1}{N} \sum_i |\phi_i - \phi_i^r|$. f_7 measures the difference between the vehicle heading and road heading. ϕ_i^r is the road heading at timestep i . This feature guarantees the vehicle's heading to be consistent with road orientation.

Efficiency Cost.

1. Speed limit: $f_8 = \frac{1}{N} \sum_i \frac{|v_i - v_{lim}|}{v_{lim}}$, where v_{lim} is the speed limit of the road. f_8 represents the efficiency of trajectory planning. The vehicle's speed is expected to be close to the speed limit of the road segment.
2. Longitudinal terminal position: $f_9 = |x_N - x_{ter}|$, where x_N is the longitudinal coordinate of the vehicle at the end of the planning horizon and x_{ter} is the longitudinal position of the terminal point.
3. Lateral terminal point: $f_{10} = |y_N - y_{ter}|$, where y_N is the lateral coordinate of the vehicle at the end of the planning horizon, and x_{ter} is the lateral position of the terminal point. f_9 and f_{10} push the vehicle to reach the desired coordinate at the end of the planning time.

Vehicle Dynamic Constraints

The vehicle dynamic constraints represent the kinematics of vehicle motion, shown in Equations 2–4. Vehicles are considered as mass points and parameters such as vehicle length and vehicle width are ignored. The constraints include the evolution of the vehicle's position, speed, and heading angle.

$$x(t+1) = x(t) + v \cos(\varphi(t)) \tau \quad (2)$$

$$y(t+1) = y(t) + v \sin(\varphi(t)) \tau \quad (3)$$

$$v(t+1) = v(t) + \tau a(t) \quad (4)$$

In addition, the boundaries of the vehicle's kinematic parameters are considered in Equations 5–9.

$$-\pi \leq \varphi(t) \leq \pi \quad (5)$$

$$-3 \leq a(t) \leq 3 \quad (6)$$

$$-0.1 \leq a(t+1) - a(t) \leq 0.1 \quad (7)$$

$$-0.2 \leq \varphi(t+1) - \varphi(t) \leq 0.2 \quad (8)$$

$$-0.15 \leq \varphi(t) - \varphi^r(t) \leq 0.15 \quad (9)$$

Equation 5 limits the vehicle's heading angle within the range $(-\pi, \pi)$. Equation 6 constrains the maximum acceleration and deceleration to be less than 3 m/s^2 , Equation 7 demonstrates that the jerk in the trajectory planning process should be less than 0.1 m/s^3 . Equation 8 sets constraints of the vehicle's heading angle change rate. Equation 9 limits the difference between the vehicle's heading angle and the road heading.

IRL

After designing all the features, the next step is to determine the associated weight vector (i.e., θ). Maximum

entropy IRL, a specific method in imitation learning, is applied to obtain the weight vector of the cost function. The IRL process is briefly reviewed below, more details can be found in Ziebart et al. (8).

First, a set of demonstration trajectories (i.e., \mathcal{D}) are needed as the training dataset. Each demonstration includes the trajectory trace of the ego vehicle and the associated driving environment parameters. The feature vector $f_i (i = 1, 2, \dots, 10)$ proposed above captures the trajectory characteristics and maps them to real values with the weight vector. The empirical feature vector of the demonstration trajectories can then be represented as

$$\tilde{f} = \frac{1}{N} \sum_{d=1}^N f(\tilde{s}_d), d \in \mathcal{D} \quad (10)$$

where \tilde{s}_d are trajectories in the demonstration dataset.

The objective of the maximum entropy IRL is to find the optimal weight vector θ that can generate trajectories similar to the demonstration trajectories. More specifically, the aim here is to find the weight vector θ that maximizes the log-likelihood function in Equation 11.

$$\theta^* = \operatorname{argmax}_{\theta} L(\theta) = \operatorname{argmax}_{\theta} \sum_{d \in \mathcal{D}} \ln P(s_d | \theta, \mathbf{u}_d) \quad (11)$$

$P(s_d | \theta, \mathbf{u}_d)$ is the probability of trajectory s_d given weight vector θ and its initial state and driving environment parameters \mathbf{u}_d . The probability is estimated based on the principle of maximum entropy and has the form

$$P(s_d | \theta, \mathbf{u}_d) = \frac{e^{-\theta^T f(s_d, \mathbf{u}_d)}}{\sum_{s_k \in \mathcal{K}_d} e^{-\theta^T f(s_k, \mathbf{u}_d)}}. \quad (12)$$

where \mathcal{K}_d is the trajectory set generated following the maximum entropy principle given the initial state and driving environment.

Therefore, the gradient of the log-likelihood function can be presented as the difference between the expected feature value calculated from the demonstrations and the expected feature value obtained from the learning model.

$$\nabla_{\theta} L(\theta) = \frac{1}{m} \sum_{d \in \mathcal{D}} E_{P(s_d | \theta, \mathbf{u}_d)} [f(s_d, \mathbf{u}_d)] - \tilde{f} \quad (13)$$

where $E_{P(s_d | \theta, \mathbf{u}_d)} [f(s_d, \mathbf{u}_d)]$ can be approximated by $f(\operatorname{argmin}_{s_d} \theta^T f(s_d, \mathbf{u}_d))$, and m is the size of the demonstration dataset.

The pseudo-code of the IRL algorithm is shown below.

Experiments

To validate the proposed trajectory planning framework, the NGSIM Lankershim Boulevard Dataset (13) is applied. This section will first introduce the data processing and then show the trajectory planning results.

Algorithm 1: Learning Weight Vector

```

Compute the feature vector value on the training dataset
 $\bar{f} = \frac{1}{N} \sum_{d=1}^N f(\hat{s}_d), d \in \mathbf{D}$ 
Initialize weight vector  $\theta$  randomly.
While  $\left\| \frac{1}{m} \sum_{d \in \mathbf{D}} E_{P(s_d|\theta, u_d)} [f(s_d, u_d)] - \bar{f} \right\| > \text{threshold} :$ 
{
  For  $i=1:m$ 
  {
    Generate vehicle trajectory  $s_i^0$  by solving Equation 1, given  $u_i \in \mathbf{D}$ 
  }
  Calculate the gradient of the log-likelihood function  $\nabla_{\theta} L(\theta)$ .
  Update feature vector  $\theta(k+1) = \theta(k) + \gamma \nabla_{\theta} L(\theta)$ ,  $\gamma$  is the
  learning rate.
}

```

Description of Dataset and Data Processing

The NGSIM Lankershim Boulevard Dataset (13) provides detailed vehicle trajectories and road geometry information. Vehicle trajectories within each road segment are extracted for this study. Because of the noise in the raw data, all trajectories are first smoothed using the Kalman filter, a widely accepted filtering method to deal with the inconsistency in vehicle trajectories (27, 28).

Figure 2 shows the comparison of the smoothed trajectory and the “real NGSIM” trajectory. As shown in Figure 2a, the smoothed trajectory is close to the real trajectory. The acceleration profile is smoother than the “real NGSIM” trajectory and the trend of the speed is close to the “real NGSIM” trajectory. It can be seen that the smoothing mainly reduces noise in speed and acceleration while maintaining vehicle position data unchanged, which is the most important factor in trajectory planning. The average Euclidean distance between the smoothed and the unsmoothed NGSIM trajectory is only 0.179 m, which essentially filters out high order noises but keeps the human driving behaviors.

Vehicle speed, acceleration, and heading angle are calculated using the smoothed trajectories. In addition to vehicle trajectories, the road geometry, which is an important feature in both in-lane driving and lane changing maneuvers, is also processed. Waypoints of the road segment are extracted from the OpenStreetMap (29) to calculate the road orientations. Each trajectory point is then mapped to the nearest road segment based on the GPS coordinates so that the difference between the heading angle and road orientation can be calculated (i.e., f_7). To obtain distances between the ego vehicle and its surrounding vehicles and calculate the safety features (i.e., f_4, f_5, f_6), the distance to stop bar of each vehicle in the road segment, position of the leading vehicle in the same lane, position of the leading vehicle in the target lane,

and position of the following vehicle in the target lane are all extracted and associated to the ego vehicle. The roadway speed limit is obtained using Google Maps. As a result, all features used for the IRL model are obtained with the processed data.

Result Analysis

Model Training. Twent-four trajectories are used for training and 98 trajectories are used for testing. In both training and testing datasets, in-lane driving trajectories and lane changing trajectories are split into half and half. The resolution of the trajectories in the dataset is 10 Hz, which is consistent with the trajectory planning model. The length of each trajectory is 6 s, to guarantee the completeness of the lane changing process.

The gradient of the log-likelihood function $\nabla_{\theta} L(\theta)$ is considered as the loss function in the IRL model. Figure 3 shows the training loss during the training process. The learning rate γ is set as 0.1. The training loss converges to 0.1 after 8 to 10 iterations.

Accuracy Evaluation. Figure 4 presents the comparison between the generated and ground truth in-lane driving trajectory. Note that the ground truth trajectory refers to the post-filtering trajectory in the NGSIM dataset. In Figure 4a, the x-axis is the longitudinal coordinate and the y-axis is the lateral coordinate. Because of the terminal cost (i.e., f_9, f_{10}), the generated trajectory ends at (almost) the same location as the ground truth trajectory within the same time period. This is an important feature for trajectory planning in the context of urban traffic control, where the travel time and travel distance of a CAV may be constrained by a central controller or consensus-based decisions. It is different from open-ended trajectory planning problems for single CAV navigation. In Figure 4, b–d, the x-axis is the simulation time and the y-axis is the speed, acceleration, and vehicle heading angle, respectively. It can be seen that the generated trajectory follows the ground truth trajectory very well and the trend for the change of speed, acceleration and heading is close to the ground truth profiles.

Figure 5 presents an example of the comparison between the generated and ground truth lane changing trajectory. As shown in Figure 5a, the position profile is also very similar to the original trajectory. The acceleration and speed profiles of the generated trajectory are smoother than the original trajectory, which motivates the comparison of fuel consumption and safety gaps discussed in the next sections.

To further quantify the performance of the model, The average position displacement in relation to Euclidean distance is used to measure the accuracy of the generated trajectory, as shown in Equation 14.

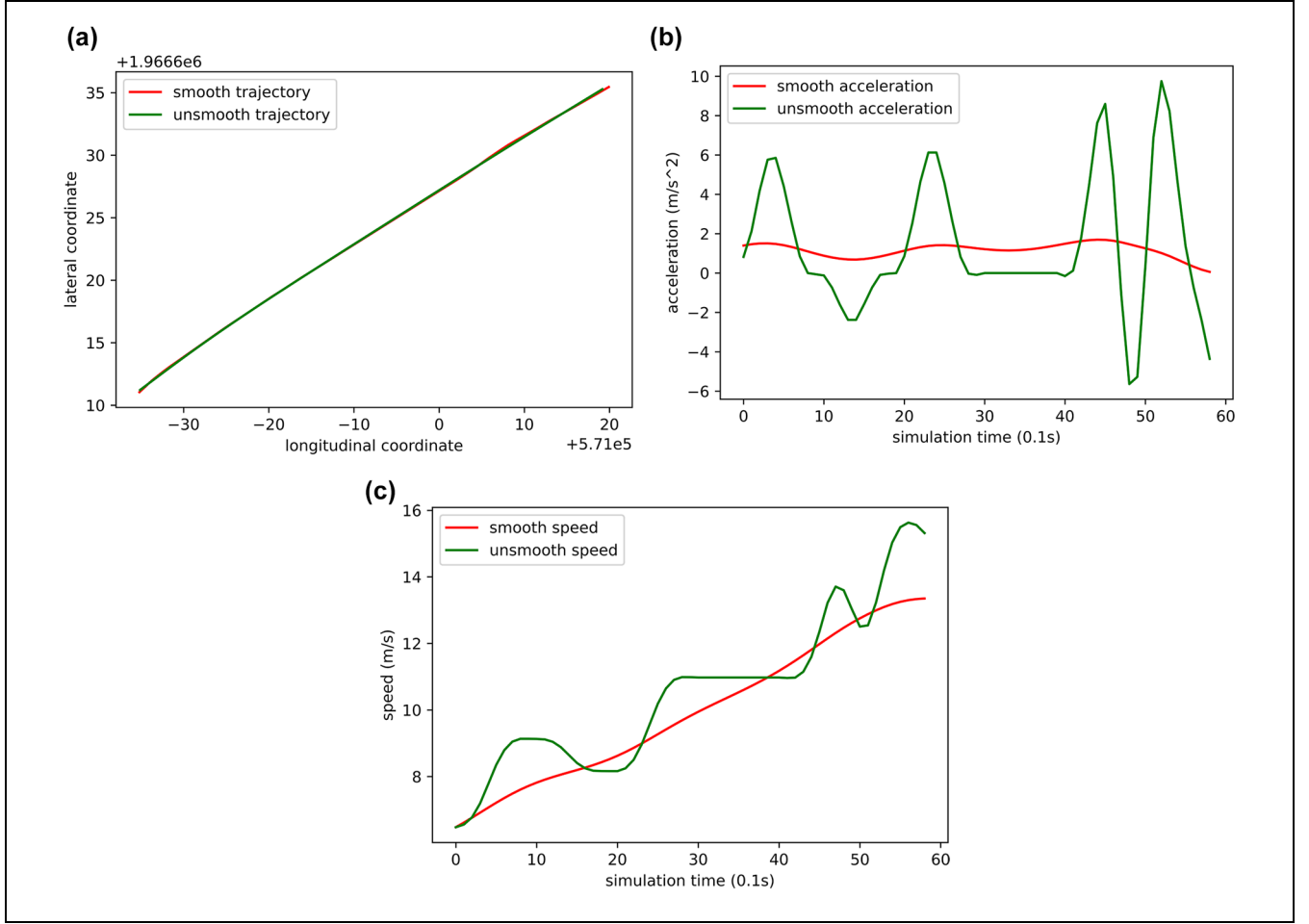


Figure 2. Comparison for smoothed trajectory and “Real NGSIM” trajectory: (a) position, (b) acceleration, and (c) speed.

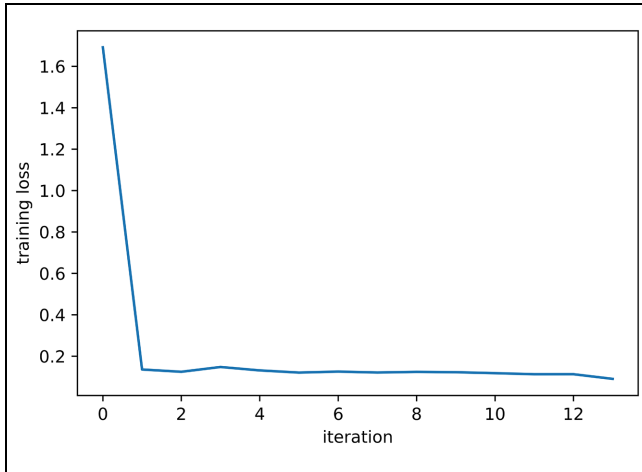


Figure 3. Training loss in inverse reinforcement learning.

$$\begin{aligned} & \text{Euclidean distance}(s_g, s_o) \\ &= \frac{\sum_{i=1}^N \sqrt{(x_{gi} - x_{oi})^2 + (y_{gi} - y_{oi})^2}}{N} \end{aligned} \quad (14)$$

where (x_{gi}, y_{gi}) is the coordinate of the generated trajectory, and (x_{oi}, y_{oi}) is the coordinate of the ground truth trajectory. N is the total length of the trajectory. In this study, $N = 60$.

Table 1 represents the average position displacement in relation to Euclidean distance for both training and testing datasets under both in-lane driving (car-following) and lane changing scenarios. Overall, the model performs quite well. The average displacement for the in-lane scenarios is less than 1 m and less than 1.2 m for the lane changing scenarios. Lane changing scenarios involve both longitudinal and lateral control, which is more difficult to imitate. Note that the length of a typical mid-sized vehicle is about 4.5 m, so an average displacement of 1 m is less than one-quarter of a car length.

Fuel Consumption. Reduced fuel consumption is another major objective in many trajectory planning models (30). Although the proposed model does not specifically include fuel consumption as part of the cost function, minimizing acceleration and jerk usually has an impact

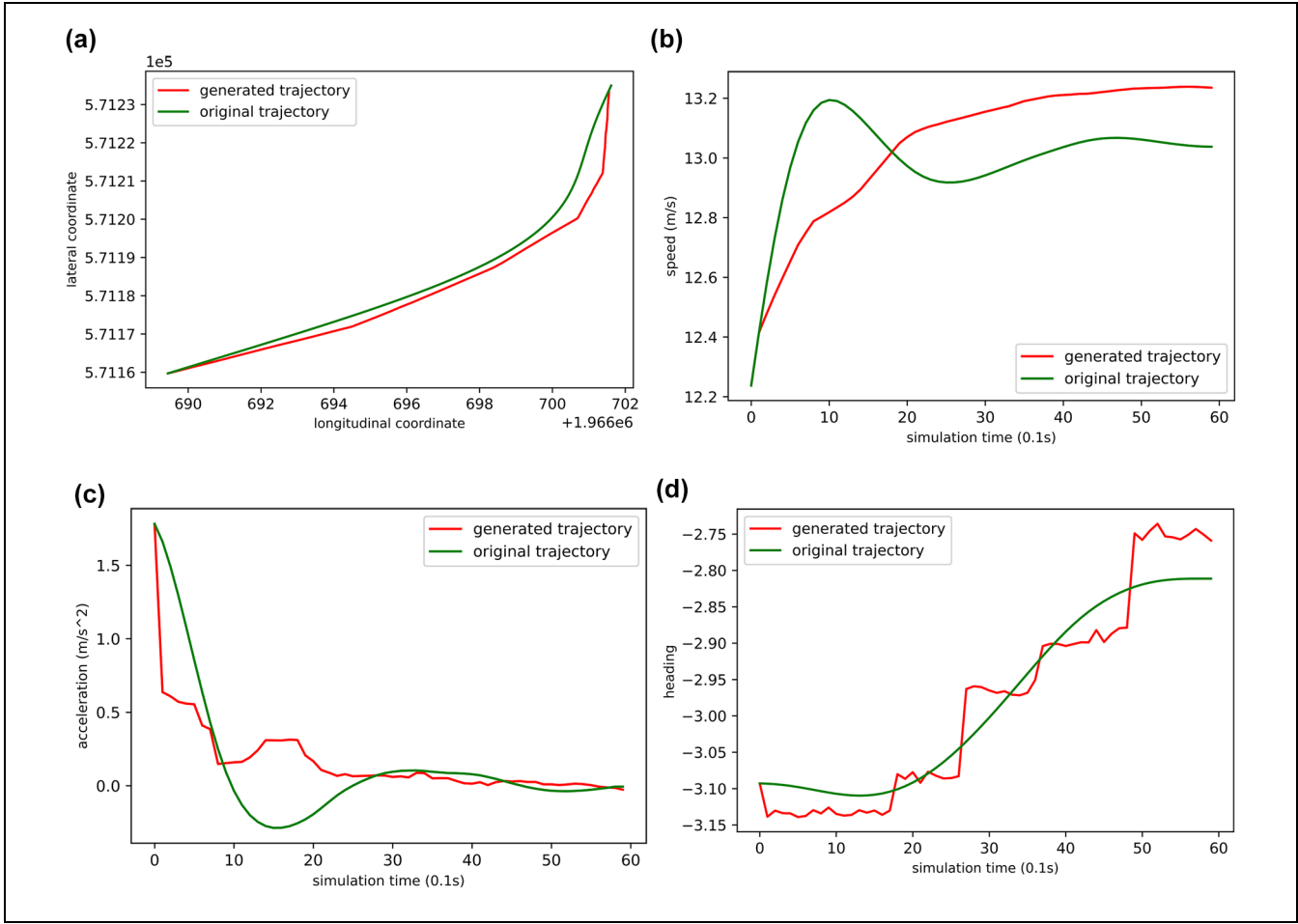


Figure 4. Comparison between ground truth and generated trajectory (in-lane): (a) position, (b) speed, (c) acceleration, and (d) heading angle.

on fuel consumption. To measure the fuel consumption of the generated trajectories quantitatively, the VT-Micro model is applied (31). The VT-Micro model measures fuel consumption and emissions as polynomial combinations of the vehicle's instantaneous speed and acceleration, as shown in the following equation.

$$\ln(MOE_e) = \begin{cases} \sum_{i=0}^3 \sum_{j=0}^3 (L_{i,j}^e \times s^i \times a^j), & \text{for } a \geq 0 \\ \sum_{i=0}^3 \sum_{j=0}^3 (M_{i,j}^e \times s^i \times a^j), & \text{for } a < 0 \end{cases} \quad (15)$$

where

$\ln(y)$ = natural logarithm function of y ;

s = instantaneous vehicle speed (km/h);

a = instantaneous vehicle acceleration (km/h/s);

MOE_e = instantaneous fuel consumption or emission rate (L/s or mg/s);

e = index denoting fuel consumption or emission type;

$M_{i,j}^e$ = model regression coefficient for MOE_e at speed power i and acceleration power j when the acceleration is negative;

$L_{i,j}^e$ = model regression coefficient for MOE_e at speed power i and acceleration power j when the acceleration is positive.

The model coefficients applied in this research to calculate the fuel consumption can be found in Alsabaan et al. (31). Table 2 shows the average fuel consumption of all generated trajectories and the comparison with the ground truth trajectories in the testing dataset under both in-lane and lane changing scenarios. Since the travel distance for each trajectory is different, the result is normalized by distance.

According to Table 2, the average fuel consumption for generated trajectories for in-lane scenarios is close to the ground truth trajectories, while the average fuel consumption for lane changing scenarios is higher by 15.9%. Although the generated trajectories for lane changing scenarios consume more fuel, the frequency of lane changing maneuvers is much less compared with in-lane driving. As a result, it is reasonable to argue that the fuel consumption of the proposed trajectory generation model is comparable with the ground truth.

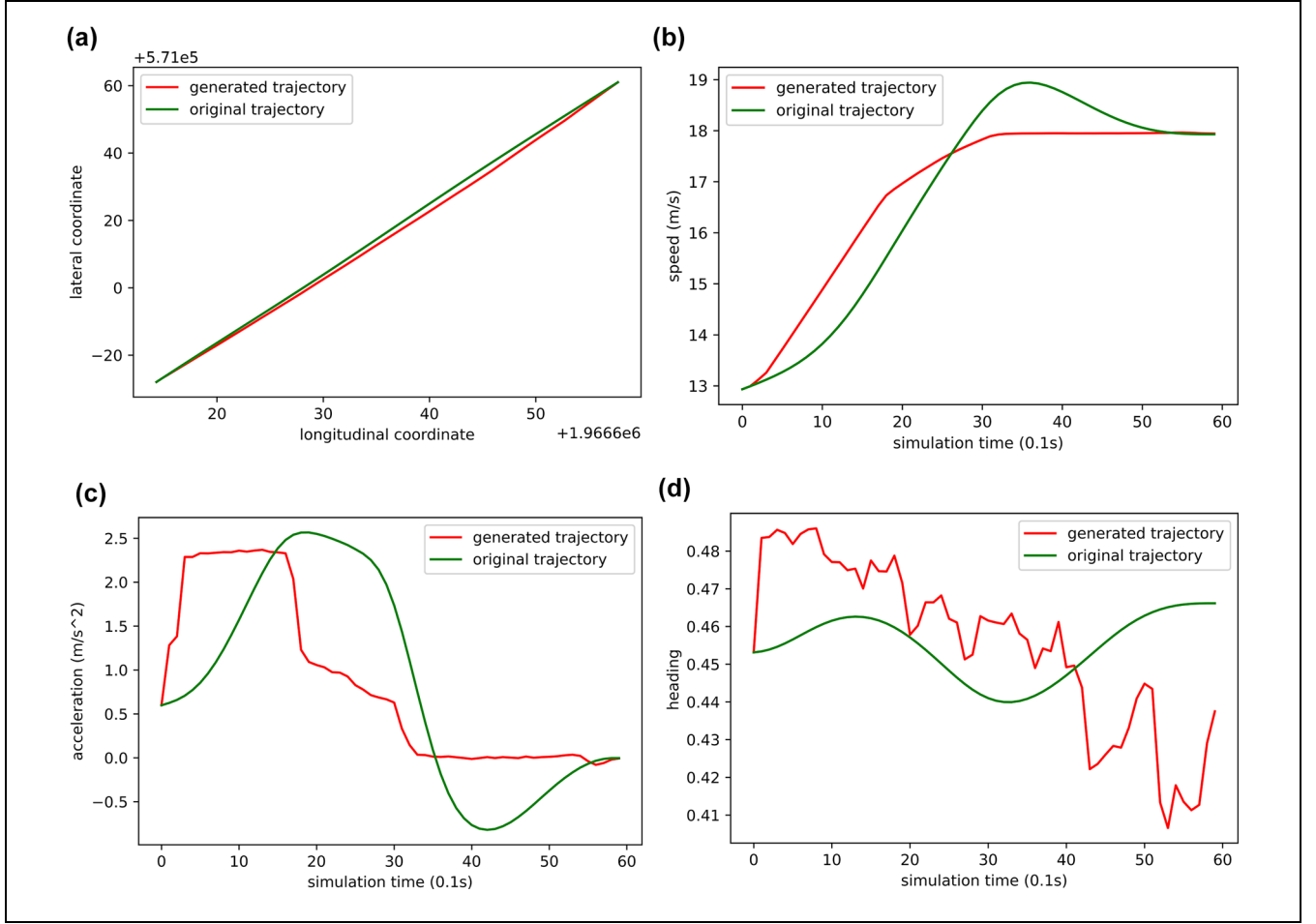


Figure 5. Comparison between ground truth and generated trajectory (lane changing): (a) position, (b) speed, (c) acceleration, and (d) heading angle.

Table 1. Euclidean Distance for Training and Testing Datasets

	Euclidean distance (training dataset) (m)	Euclidean distance (testing dataset) (m)
In-lane	0.831	0.834
Lane changing	1.228	1.184
All	1.005	0.997

Table 2 Fuel Consumption for Original and Generated Trajectory

	Ground truth trajectory (mL/m)	Generated trajectory (mL/m)
In-lane	1.222	1.294
Lane changing	1.760	2.040

Safety Performance. Safety is the most critical criterion in trajectory planning. To validate whether the generated trajectories satisfy the safety gaps between the ego vehicle and its critical surrounding vehicles, the distributions of gaps during the planning period between the generated trajectories and ground truth trajectories are compared. Three types of critical surrounding vehicles are considered. For in-lane driving scenarios, the gap to the leading vehicle is considered critical. For lane changing scenarios, in addition to the leading vehicle in the same lane, the leading and following vehicle in the target lane

are also considered critical. Note that, after the lane changing is complete, only the leading vehicle in the same lane is considered. Figure 6 shows an example of the gap distributions in a lane changing scenario. Figure 6a presents the gap distributions of the leading vehicle distance. Figure 6, b and c, are the gap distributions of the leading and following vehicles in the target lane. The x-axis is the distance to the surrounding vehicle and the y-axis is the frequency. Although the distance distribution of the generated trajectory is slightly different from the ground truth trajectory, the safety gaps for the

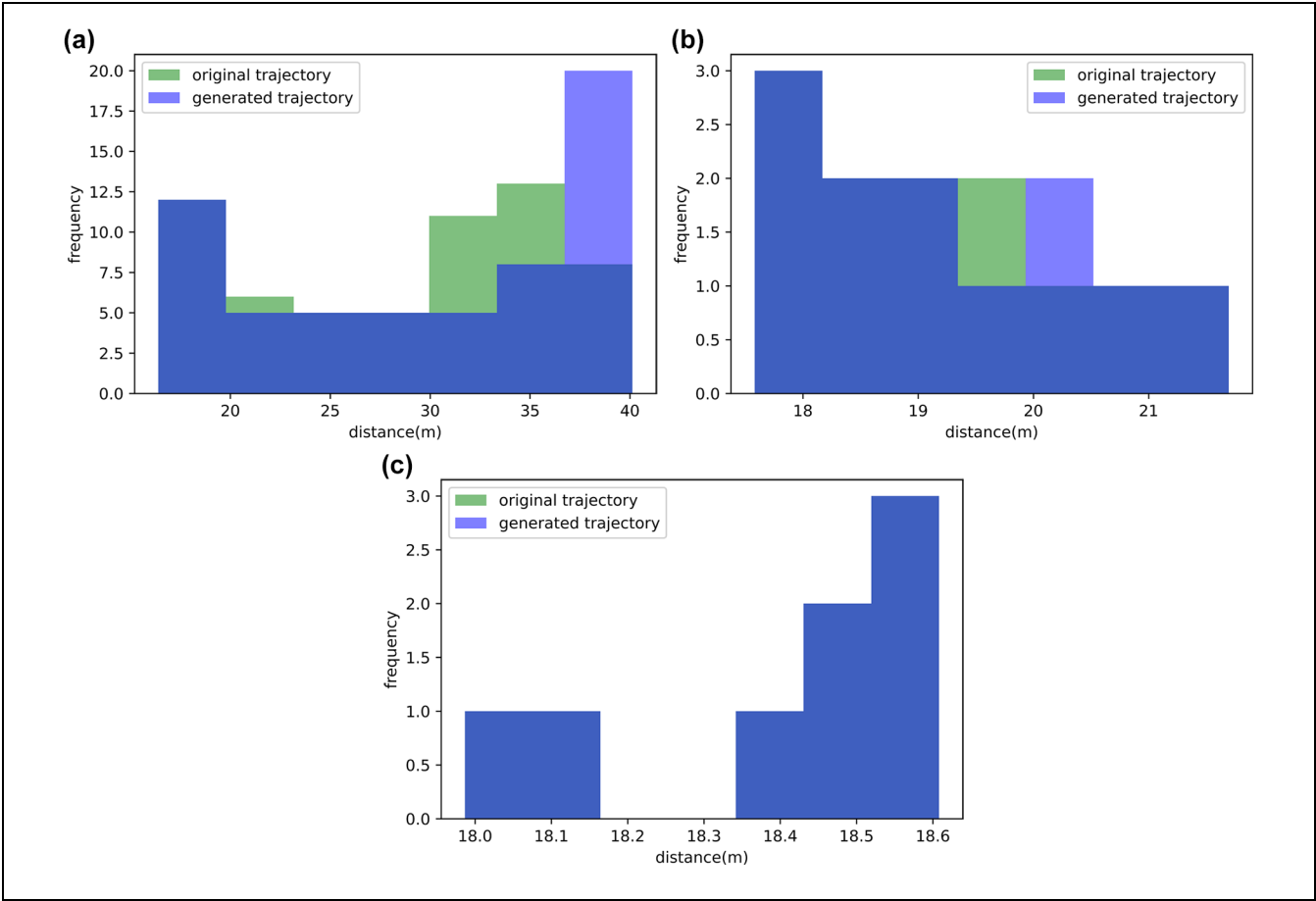


Figure 6. Safety gap distribution comparison: (a) front vehicle distance distribution, (b) target lane front vehicle distance distribution, and (c) target lane following vehicle distance distribution.

generated trajectory are no smaller than those of the ground truth trajectory, which illustrates that generated trajectories do not present more aggressive driving behaviors.

To further quantify critical gaps of all generated trajectories, mean average error (MAE) is applied to measure the difference between the gap distributions. The MAE is calculated by Equation 16, which measures the difference in safety gaps between the ground truth and generated trajectories.

$$MAE(P, Q) = \frac{1}{m} \sum_{i=1}^m |p_i - q_i| \quad (16)$$

where p_i is the safety gap for generated trajectory P at time step i , q_i is the safety gap for the original trajectory Q at time step i . m is the total number of time steps when the leading vehicle in the same lane, the leading vehicle in the target lane, and the following vehicle in the target lane are all present.

MAEs for all trajectories are aggregated to reflect the overall performance. Table 3 shows that the average

Table 3. Mean Average Error (MAE) and Standard Deviation of Safety Gaps between Original and Generated Trajectories

	MAE (m)	Standard deviation (m)
Leading vehicle (same lane)	0.604	0.507
Leading vehicle (target lane)	0.711	0.736
Lagging vehicle (target lane)	0.706	0.711

MAE of gaps for all critical surrounding vehicles is less than 0.75 m and the standard deviation is less than 0.75 m. The results further validate that the proposed model has very similar in-lane and lane changing behaviors to the ground truth trajectories.

Conclusion, Discussion, and Future Research

In this research, an imitation learning model to learn human driving policies for full vehicle trajectory

planning is proposed. Maximum entropy IRL was applied to obtain the parameters of the cost function. The trajectory generation problem was then formulated as an optimization problem with the parameters obtained from the IRL model. In the numerical study, a real-world dataset, the NGSIM Lankershim Boulevard Dataset, was applied to evaluate the model performance. Distance displacement in relation to Euclidean distance is calculated to measure the accuracy of the generated trajectories. Results show that the average displacement is only about one-quarter of a passenger vehicle's length, which indicates that the generated trajectory can mimic human driving behavior very well. Meanwhile, the safety gaps between the CAVs with generated trajectories and their critical surrounding vehicles are guaranteed and the fuel consumption is comparable with that of the ground truth trajectories.

The proposed model could be applied for urban traffic control, especially in a cooperative driving environment where traffic signals and vehicle trajectories are jointly optimized. Hierarchical control frameworks are usually applied in the joint control problem where the infrastructure provides high-level guidance based on trajectory data collected from CAVs while each CAV plans its own trajectory independently following the guidance (5, 32). This architecture has advantages over purely centralized control where the dimensionality of the problem is huge and purely distributed control where the system optima may not be reached. The proposed model can fit well into the hierarchical control framework. The high-level guidance from the infrastructure can be converted to the terminal coordinates in the cost function that forces the vehicle to reach the destination on time. The planned trajectory and observed trajectories from surrounding vehicles can be sent to the infrastructure to serve as input data for future traffic control decisions. There are two benefits to generating CAV trajectories that mimic human driving behaviors. First, it simplifies the traffic flow model used to predict traffic states at the infrastructure side. Instead of mixed traffic conditions with heterogeneous traffic, a homogeneous traffic flow model with one vehicle type can be applied. Second, some studies show that human driving behaviors may change around the CAVs, depending on their level of trust in the technology (33). A human-like CAV trajectory could increase the trust level and decrease the variations in human driving behaviors.

In future work, the authors will investigate how to incorporate the impact of traffic signals into the imitation learning trajectory planning framework. The authors also plan to integrate the proposed trajectory planning model into a cooperative driving framework, such as integrated optimization with traffic signals. In addition, it would be interesting to explore further the reason why

lane changing scenarios consume more fuel, using more detailed fuel consumption and vehicle powertrain models.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: J. Ying, Y. Feng; data collection: J. Ying; analysis and interpretation of results: J. Ying, Y. Feng; draft manuscript preparation: J. Ying, Y. Feng. All authors reviewed the results and approved the final version of the manuscript.


Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is supported in part by the U.S. National Science Foundation through grants SaTC #1930041 and CPS #2038215.

ORCID iD

Yiheng Feng  <https://orcid.org/0000-0001-5656-3222>

References

1. Cheng, Y., C. Chen, X. Hu, K. Chen, Q. Tang, and Y. Song. Enhancing Mixed Traffic Flow Safety via Connected and Autonomous Vehicle Trajectory Planning with a Reinforcement Learning Approach. *Journal of Advanced Transportation*, Vol. 2021, 2021, p. e6117890.
2. Du, R., S. Chen, Y. Li, J. Dong, P. Y. J. Ha, and S. Labi. A Cooperative Control Framework for CAV Lane Change in a Mixed Traffic Environment. *arXiv Preprint arXiv:201005439*, 2020. <http://arxiv.org/abs/2010.05439>. Accessed 29 July 2021.
3. Ma, M., and Z. Li. A Time-Independent Trajectory Optimization Approach for Connected and Autonomous Vehicles Under Reservation-Based Intersection Control. *Transportation Research Interdisciplinary Perspectives*, Vol. 9, 2021, p. 100312.
4. Fassbender, D., B. C. Heinrich, T. Luettel, and H. -J. Wuensche. An Optimization Approach to Trajectory Generation for Autonomous Vehicle Following. *Proc., 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 3675–3680.
5. Yang, Z., Y. Feng, and H. X. Liu. A Cooperative Driving Framework for Urban Arterials in Mixed Traffic Conditions. *Transportation Research Part C: Emerging Technologies*, Vol. 124, 2021, p. 102918.
6. Wang, P., D. Liu, J. Chen, H. Li, and C. -Y. Chan. Decision Making for Autonomous Driving via Augmented Adversarial Inverse Reinforcement Learning. *arXiv*

- Preprint *arXiv:191108044*, 2021. <http://arxiv.org/abs/1911.08044>. Accessed July 30, 2021.
7. Kuderer, M., S. Gulati, and W. Burgard. Learning Driving Styles for Autonomous Vehicles from Demonstration. *Proc., 2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, 2015, pp. 2641–2646.
 8. Ziebart, B. D., A. Maas, J. A. Bagnell, and A. K. Dey. Maximum Entropy Inverse Reinforcement Learning. *Proc., 23rd AAAI Conference on Artificial Intelligence*, Chicago, IL, 2008.
 9. Bojarski, M., D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, et al. End to End Learning for Self-Driving Cars. *arXiv Preprint arXiv:160407316*, 2016. <http://arxiv.org/abs/1604.07316>. Accessed July 30, 2021.
 10. Zhang, J., and K. Cho. Query-Efficient Imitation Learning for End-to-End Autonomous Driving. *arXiv Preprint arXiv:160506450*, 2016. <http://arxiv.org/abs/1605.06450>. Accessed July 30, 2021.
 11. Yang, D., S. Zheng, C. Wen, P. J. Jin, and B. Ran. A Dynamic Lane-Changing Trajectory Planning Model for Automated Vehicles. *Transportation Research Part C: Emerging Technologies*, Vol. 95, 2018, pp. 228–247.
 12. Xie, J., Z. Shao, Y. Li, Y. Guan, and J. Tan. Deep Reinforcement Learning with Optimized Reward Functions for Robotic Trajectory Planning. *IEEE Access*, Vol. 7, 2019, pp. 105669–105679.
 13. Lankershim Boulevard Dataset, FHWA-HRT-07-029. Federal Highway Administration. [Internet]. <https://www.fhwa.dot.gov/publications/research/operations/07029/index.cfm>. Accessed July 30, 2021.
 14. Mahmassani, H. S. 50th Anniversary Invited Article—Autonomous Vehicles and Connected Vehicle Systems: Flow and Operations Considerations. *Transportation Science*, Vol. 50, No. 4, 2016, pp. 1140–1162.
 15. Zhou, H., J. A. Laval, A. Zhou, Y. Wang, W. Wu, Z. Qing, and S. Peeta. Review of Learning-Based Longitudinal Motion Planning for Autonomous Vehicles: Research Gaps Between Self-Driving and Traffic Congestion. *Transportation Research Record: Journal of the Transportation Research Board*, 2021. 2676: 324–341.
 16. Kesting, A., M. Treiber, and D. Helbing. General Lane-Changing Model MOBIL for Car-Following Models. *Transportation Research Record: Journal of the Transportation Research Board*, 2007. 1999: 86–94.
 17. Talebpour, A., H. S. Mahmassani, and S. H. Hamdar. Modeling Lane-Changing Behavior in a Connected Environment: A Game Theory Approach. *Transportation Research Procedia*, Vol. 7, 2015, pp. 420–440.
 18. Ahmed, K. I. *Modeling Drivers' Acceleration and Lane Changing Behavior*. Thesis. Massachusetts Institute of Technology, Cambridge, 1999. <https://dspace.mit.edu/handle/1721.1/9662>. Accessed July 30, 2021.
 19. Zhang, S., W. Deng, Q. Zhao, H. Sun, and B. Litkouhi. Dynamic Trajectory Planning for Vehicle Autonomous Driving. *Proc., 2013 IEEE International Conference on Systems, Man, and Cybernetics*, Manchester, 2013, pp. 4161–4166.
 20. Luo, Y., Y. Xiang, K. Cao, and K. Li. A Dynamic Automated Lane Change Maneuver Based on Vehicle-to-Vehicle Communication. *Transportation Research Part C: Emerging Technologies*, Vol. 62, 2016, pp. 87–102.
 21. Yu, C., Y. Feng, H. X. Liu, W. Ma, and X. Yang. Integrated Optimization of Traffic Signals and Vehicle Trajectories at Isolated Urban Intersections. *Transportation Research Part B: Methodological*, Vol. 112, 2018, pp. 89–112.
 22. Yu, C., Y. Feng, H. X. Liu, W. Ma, and X. Yang. Corridor Level Cooperative Trajectory Optimization with Connected and Automated Vehicles. *Transportation Research Part C: Emerging Technologies*, Vol. 105, 2019, pp. 405–421.
 23. Naumann, M., L. Sun, W. Zhan, and M. Tomizuka. Analyzing the Suitability of Cost Functions for Explaining and Imitating Human Driving Behavior Based on Inverse Reinforcement Learning. *Proc., 2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 5481–5487.
 24. Andersen, H., W. Schwarting, F. Naser, Y. H. Eng, M. H. Ang, D. Rus, and J. Alonso-Mora. Trajectory Optimization for Autonomous Overtaking with Visibility Maximization. *Proc., 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Japan, 2017, pp. 1–8.
 25. Best, A., S. Narang, D. Barber, and D. Manocha. AutoVi: Autonomous Vehicle Planning with Dynamic Maneuvers and Traffic Constraints. *Proc., 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 2629–2636.
 26. Sharifzadeh, S., I. Chiotellis, R. Triebel, and D. Cremers. Learning to Drive Using Inverse Reinforcement Learning and Deep Q-Networks. *arXiv Preprint arXiv:161203653*, 2017. <http://arxiv.org/abs/1612.03653>. Accessed November 24, 2021.
 27. Punzo, V., D. J. Formisano, and V. Torrieri. Nonstationary Kalman Filter for Estimation of Accurate and Consistent Car-Following Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2005. 1934: 2–12.
 28. Ma, X., and I. Andreasson. Dynamic Car Following Data Collection and Noise Cancellation Based on the Kalman Smoothing. *Proc., IEEE International Conference on Vehicular Electronics and Safety*, Shaanxi, China, Vol. 2005, 2005, pp. 35–41.
 29. Haklay, M., and P. Weber. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, Vol. 7, No. 4, 2008, pp. 12–18.
 30. Liu, X., G. Zhao, N. Masoud, and Q. Zhu. Trajectory Planning for Connected and Automated Vehicles: Cruising, Lane Changing, and Platooning. *arXiv Preprint arXiv:200108620*, 2020. <http://arxiv.org/abs/2001.08620>. Accessed July 30, 2021.
 31. Alsabaan, M., K. Naik, T. Khalifa, and A. Nayak. Applying Vehicular Networks for Reduced Vehicle Fuel Consumption and CO₂ Emissions. In *Intelligent Transportation Systems* (Abdel-Rahim, A., ed.), InTechOpen, London,

2012. <https://www.intechopen.com/chapters/32777>. Accessed January 29, 2022.
32. Chalaki, B., and A. A. Malikopoulos. Optimal Control of Connected and Automated Vehicles at Multiple Adjacent Intersections. *IEEE Transactions on Control Systems Technology*, 2021, pp. 1–13.
33. Zhao, X., Z. Wang, Z. Xu, Y. Wang, X. Li, and X. Qu. Field Experiments on Longitudinal Characteristics of

Human Driver Behavior Following an Autonomous Vehicle. *Transportation Research Part C: Emerging Technologies*, Vol. 114, 2020, pp. 205–224.

The views presented in this paper are those of the authors alone. This paper does not constitute a standard, specification, or regulation.