2024

Modeling Coupled Driving Behavior During Lane Change: A Multi-Agent Transformer Reinforcement Learning Approach

Hongyu Guo
University of Canterbury

Mehdi Keyvan-Ekbatani
University of Canterbury

Kun Xie
Old Dominion University, kxie@odu.edu

Original Publication Citation

Guo, H., Keyvan-Ekbatani, M., & Xie, K. (2024). Modeling coupled driving behavior during lane change: A multi-agent Transformer reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, *165*, 1-19, Article 104703. https://doi.org/10.1016/j.trc.2024.104703

Transportation Research Part C 165 (2024) 104703

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Modeling coupled driving behavior during lane change: A multi-agent Transformer reinforcement learning approach ☆

Hongyu Guo [a], Mehdi Keyvan-Ekbatani [a,*], Kun Xie [b]

[a] Complex Transport Systems Laboratory (CTSLAB), Department of Civil and Natural Resources Engineering, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand

[b] Transportation Informatics Lab, Department of Civil and Environmental Engineering, Old Dominion University, Norfolk, VA 23529, United States

ARTICLE INFO

Keywords:
Lane change modelling
Multi-agent reinforcement learning
Transformer reinforcement learning
Connected vehicle
Big data analytics

ABSTRACT

In a lane change (LC) scenario, the lane change vehicle interacts with surrounding vehicles. The interactions not only affect their driving behaviors but also influence the traffic flow. This study aims to model the coupled behavior of the lane changer and the follower in the target lane during LC. Large-scale real-world connected vehicle (CV) data from the Safety Pilot Model Deployment (SPMD) program are used to extract LCs and study vehicle interactions. A multi-agent Transformer-based deep deterministic policy gradient (MA-TDDPG) method is proposed to model the coupled behaviors during LC. The multi-agent framework can handle the multiple agents' behaviors with interactions, and the Transformer can process the observation-action memory accurately and efficiently. The MA-TDDPG algorithm can learn the sequential decision-making process over continuous action space during LC with the accommodation of the multi-vehicle interaction and the driver's memory effect. Compared to traditional supervised learning and reinforcement learning methods, it demonstrates superior performance in imitating the longitudinal and lateral actions of the lane changer and the follower. The findings of this study provide insights into the development of microscopic simulations by producing realistic LC behaviors, and assistance/automation LC systems by generating LC motions and responses conforming to human driving habits. The model also creates an interactive simulation environment and lays the foundation for optimizing driving strategies.

1. Introduction

Lane change (LC) is a complicated driving behavior that depends on the surrounding traffic dynamics, lane positions, and driver's motivations (Sun and Kondyli, 2010; Keyvan-Ekbatani et al., 2016a; Ali et al., 2020; Zhang et al., 2021). A typical LC process involves the lane changer and its surrounding vehicles. Improper vehicle interactions during LC can cause negative impacts on traffic flow and safety (Ahn and Cassidy, 2007; Zheng et al., 2010, 2011). Thus, it is vital to analyze and model the cooperative and competitive driving behaviors of multiple vehicles involved in an LC. The accurate modeling could contribute to the microscopic traffic simulation by generating more realistic LC trajectories (Zheng, 2014; Keyvan-Ekbatani et al., 2016b). It is also beneficial for the development of the advanced driving assistance system (ADAS) to execute LC or respond to LC consistent with human operating habits (Bevly et al., 2016).

☆ This article belongs to the Virtual Special Issue on IG005593: VSI:MLCATS.

* Corresponding author.
E-mail addresses: hongyu.guo@pg.canterbury.ac.nz (H. Guo), mehdi.ekbatani@canterbury.ac.nz (M. Keyvan-Ekbatani), kxie@odu.edu (K. Xie).

https://doi.org/10.1016/j.trc.2024.104703
Received 20 September 2023; Received in revised form 30 April 2024; Accepted 5 June 2024
Available online 14 June 2024

Analyzing and modeling LC behavior has always been an area of research interest because of its important role in traffic dynamics. Aside from LC decision models (Gipps, 1986; Toledo et al., 2003; Kesting et al., 2007), significant progress has been made in modeling the lane changer's behavior during LC execution (Xie et al., 2019; Zhang et al., 2019; Zhao et al., 2020) and the LC impact on follower's behavior (Laval and Leclercq, 2008; Duret et al., 2011; Zheng et al., 2013). However, LC is often considered a one-way process, and the interaction between the lane changer and surrounding vehicles is generally ignored. The interplay could play an important role in the LC process (Kita, 1999; Yu et al., 2018; Ali et al., 2020), and the ignorance could lead to biased LC models (Ali et al., 2019). Thus this study focuses on coupled LC behavior modeling by accommodating vehicle interactions.

Meanwhile, reinforcement learning (RL) algorithms have been used in LC modeling, which is expected to capture the underlying strategy and imitate the driver's behaviors sequentially (Aradi, 2020). It can be applied in both upper-level LC decision making (Ye et al., 2019; Li et al., 2022b) and the lower-level LC motion control (Wang et al., 2019b; Yu et al., 2022). Although the following vehicle's reward was included in some studies (Wang et al., 2021a; Jiang et al., 2022), its behavior has not been modeled properly using RL. Multi-agent RL (MA-RL) is a type of reinforcement learning, which could model the behaviors of multiple agents with interactions (Busoniu et al., 2008). Besides, the environment is treated as a Markov decision process (MDP) in conventional RL models and the agent's decision is made based on the single-step state. It might be inappropriate given that the vehicle actions are not memoryless (Zhang, 2003). Recurrent RL (RRL) provides a solution to incorporate the memory effect in the RL algorithm by implementing a module to process sequential data. It is worth further unlocking the potential of MA-RL and RRL approaches in LC behavior modeling.

Previous studies on LC behavior modeling commonly used vehicle trajectory data (e.g., NGSIM) and simulation data, whereas the former only covers limited driving scenarios and the latter cannot reflect realistic driving behavior. The connected vehicle (CV) equipped with onboard devices could collect the naturalistic driving data of itself and surrounding vehicles. The large-scale real-world data connected by CVs are beneficial for LC behavior modeling (Toledo, 2007; Zheng, 2014). The Safety Pilot Model Deployment (SPMD) program is the world's largest CV test program conducted by the U.S. Department of Transportation (USDOT) (Henclewood et al., 2014). The rich high-resolution data have shown great potential for LC research (Zhao et al., 2017; Guo et al., 2021, 2022). They also create the opportunity to model the interactions during the LC process under various driving scenarios.

This study aims to develop a coupled LC behavior model of the LC vehicle and the following vehicle in the target lane. The multi-agent Transformer-based deep deterministic policy gradient (MA-TDDPG) method is developed to model the coupled driving behavior. The multi-agent framework is adopted to imitate the cooperative and/or competitive interactions between vehicles, and the Transformer network can incorporate the memory effect of drivers. The large-scale real-world driving data collected by CVs in the SPMD program is used for model development. This study contributes to the literature in the following aspects.

- The driving behaviors of the LC vehicle and the following vehicle are modeled jointly.
- The MA-TDDPG method is proposed to model the coupled LC behavior considering multi-vehicle interaction and memory effect.
- The large-scale naturalistic driving data collected by CVs are used for model development.

The remainder of this paper is organized as follows. The related works are reviewed in Section 2. The MA-TDDPG LC model is introduced in Section 3. The detailed CV data processing procedure is illustrated in Section 4. The results and comparisons are presented in Section 5. At last, the conclusions of this study are summarized in Section 6.

2. Literature review

2.1. Driving behaviors of lane changer and follower

Different from the extensively studied LC decision (Gipps, 1986; Hidas, 2002; Toledo et al., 2003; Kesting et al., 2007) and LC prediction models (Kumar et al., 2013; Xing et al., 2020; Wang et al., 2021b; Guo et al., 2022), only a few models are proposed to capture the LC dynamics and its impacts on surrounding vehicles. Some studies used mathematical functions to describe the trajectory of LC vehicle (Yao et al., 2013; Butakov and Ioannou, 2014; Zhou et al., 2017; Zhao et al., 2020). These models could provide smoothing LC trajectories consistent with the real LC data, although they might not have clear physical meanings.

Recently, with the rapid development of data collection technologies and deep learning approaches, many studies utilized data-driven methods to model the LC trajectory. Xie et al. (2019) proposed an LC implementation model based on the long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997), which was able to predict the LC trajectory accurately. Zhang et al. (2019) considered the car following and LC behaviors in one framework and developed a hybrid retraining-constrained LSTM model to simulate them simultaneously. Similarly, Shi et al. (2022) proposed an integrated model based on the attention mechanism and temporal convolution network. Based on the LC segmentation result (Zheng et al., 2013; Chen et al., 2021) proposed LSTM models with different inputs to predict the LC trajectory at different stages. Wei et al. (2022a) presented a heuristic model based on the attention-aided encoder–decoder structure to predict the LC vehicle's trajectory and kinematics. This kind of data-driven method is also known as clone learning, which could clone the driver's LC behavior and achieve higher accuracy with a longer prediction horizon compared to the mathematical models.

In the research about LC impact, the LC process is usually divided into two stages, namely anticipation and relaxation (Zheng et al., 2013). The anticipation starts when the lane changer initiates a maneuver in the original lane. The relaxation refers to

the process in which the lane changer adjusts the short spacing upon the LC maneuver to the normal spacing given the speed. Some studies evaluated the relaxation phenomenon and its influence in a car following framework. Laval and Leclercq (2008) incorporated a macroscopic LC model (Laval and Daganzo, 2006) and proposed a microscopic modeling framework to describe the relaxation phenomenon. The model was verified by Leclercq et al. (2007) using the NGSIM data (Alexiadis et al., 2004) and found to be consistent with macroscopic observations. The model was then reformulated by using the microscopic maximum passing rate instead of the macroscopic backward-moving kinematic wave speed (Duret et al., 2011). However, these studies only focused on the transition process after the lane changer appears in the target lane. The LC impact likely begins from the anticipation period.

To investigate the impact during the entire LC transition process, research based on Newell's car following model (Newell, 2002) has been done to examine the LC effects (Wang and Coifman, 2008; Ma and Ahn, 2008). By investigating the speed-spacing relations under car following and LC conditions, it is reported that the LC maneuver would affect the lane changer, the follower in the initial lane, and the follower in the target lane during anticipation and relaxation periods. Zheng et al. (2013) further extended the aforementioned model (Laval and Leclercq, 2008; Duret et al., 2011) to describe both the anticipation and relaxation processes simultaneously. It was found that the follower's trajectory was more correlated with the lane changer rather than the leader during the anticipation period.

However, a gap existing in these studies is that the driving behavior of the lane changer and follower are considered separately. Only the lane changer's or follower's maneuver is modeled, and the other vehicle would follow a fixed trajectory according to the data. In fact, their driving behaviors would affect each other, as evidenced by the widely used game theory-based LC models (Kita, 1999; Yu et al., 2018; Ali et al., 2019; Wang et al., 2022a). The interaction between the lane changer and follower is ignored in the modeling of LC dynamics and impacts. It is necessary to develop a model for the coupled behaviors of both the lane changer and follower.

Besides, the complete vehicle trajectory data are widely used in these studies. This kind of data collected by stationary equipment can cover a limited range of road segments, where the traffic conditions are similar. The varied LC scenarios are only partly covered in the data. Thus the mathematical or clone learning models might not be general enough to be applied in a different environment. In contrast, CVs can collect various LC scenarios under multiple traffic conditions, which is beneficial for LC modeling (Toledo, 2007; Zheng, 2014). The RL model is expected to explore the underlying strategies of driving behaviors rather than simple imitation, which could have better transferability. So it is worth developing an RL-based LC model using the CV data.

2.2. Reinforcement learning in driving behavior modeling

Reinforcement learning (RL) is a machine learning approach aiming to let an intelligent agent interact with an environment and make decisions. The agent is trained to learn an optimal policy that maximizes the reward function. The basic RL is modeled as a Markov decision process (MDP). At each time step t, the RL agent receives a state S_t. It chooses an action A_t according to the policy $\pi(A_t|S_t)$, which is subsequently sent to the environment. Then, the environment moves to a new state S_{t+1} and gives a reward R_t to the agent. The MDP continues until the system reaches a terminal state and it will restart. The agent aims to maximize the discounted accumulated reward function $\sum R_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$, with the discount factor $\gamma \in (0, 1]$ (Li, 2017). As the RL algorithm improves its policies by interacting with the environment, it is suitable for problems whose solutions can be optimized by trial-and-error. It is also appropriate for problems that emphasize task completion and delayed reward over the periodical success at intermediate steps. RL was widely applied in transportation studies and provided promising results (Ozan et al., 2015; Aslani et al., 2017; Essa and Sayed, 2020; Guo et al., 2023; Jiang et al., 2024).

By combining RL and deep learning (DL), the deep reinforcement learning (DRL) can be formulated, whose policy and other learned functions are often represented as neural networks. As DL is incorporated into the RL solution, the agent is able to make decisions from complex high-dimensional raw input data with less manual engineering of the state space. Due to the ability to generate high-quality solutions and generality in solving varying problems, DRL has been widely used in driving behavior modeling aiming at the application in autonomous vehicles (AVs) (Haydari and Yilmaz, 2020; Farazi et al., 2021). As for LC-related studies, the DRL in discrete action space was used for upper-level LC decision modeling (Ye et al., 2019; Dong et al., 2021). Based on the deep Q-Network (DQN), Wang et al. (2019a) included the rule-based constraints to achieve a safe and efficient LC. Li et al. (2022b) proposed a risk-aware RL model to find the LC decision with the minimum expected risk. Similarly, He et al. (2022) focused on LC safety and developed an observation adversarial RL approach to ensure safety under perception uncertainty. The DRL in continuous action space has been applied for lower-level LC motion control. Wang et al. (2019b) proposed a deep deterministic policy gradient (DDPG) (Lillicrap et al., 2015) model as the lateral controller to decide the vehicle's yaw acceleration. In cooperation with the intelligent driver model (IDM) (Treiber et al., 2000) as the longitudinal controller, the LC behavior can be modeled. Chen et al. (2019) developed a hierarchical LC behavior model based on DDPG, whose action space consists of both decision making and route planning. Yu et al. (2022) used DDPG to model both the longitudinal and lateral acceleration during LC, and utilized NGSIM data for model training and testing. The DRL algorithms were also widely used in CF modeling (Zhu et al., 2018; Lin et al., 2019b; Zhu et al., 2020). Compared to clone learning, RL is expected to achieve better generalizability and transferability. Because RL models could infer and learn the inherent strategies rather than imitate the driver's behaviors simply.

Although the cooperative factors were included through reward functions in some studies (Wang et al., 2021a; Jiang et al., 2022; Wang et al., 2022b), the interactions between vehicles were not properly modeled in these single-agent models. Besides, the action was decided according to a single-step state, whose information might be not enough to represent the complete state. To bridge these two gaps, MA-RL (Lowe et al., 2017) and recurrent RL (RRL) (Heess et al., 2015) can be adopted for driving behavior modeling.

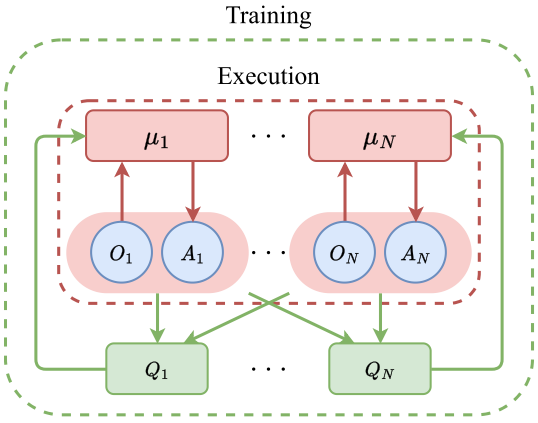


Fig. 1. Overview of MA-DDPG structure (Lowe et al., 2017).

As two extensions of the basic RL algorithm, MA-RL incorporates the interactions of multiple agents in a common environment, and RRL adds the history of previous states and actions to the intelligent agent. They are suitable for developing models considering the influence of surrounding vehicles and driver's memories. But they were rarely used in driving behavior modeling. Alsaleh and Sayed (2021) used MA-RL to simulate the cyclist-pedestrian interactions in shared spaces. Chen et al. (2019) added recurrence to the DDPG-based LC model to generate more complex driving strategies with the ability to handle the longer sequence input. Wang and Chan (2017) applied LSTM to capture the historical driving information and conveyed it to DQN for selecting the on-ramp merging action. Peake et al. (2020) used LSTM network and connectionist RL (Williams, 1992) to achieve the cooperative adaptive cruise control.

In LC studies, Zhang et al. (2022) developed a collaborative LC decision model with MA-RL. Li et al. (2022a) used the Transformer to process sequential images in DQN and proposed an LC decision model. To the best of our knowledge, the complete multi-agent recurrent RL (MA-RRL) has only been used in traffic signal control (Wu et al., 2020). This is one of the first studies that explore the potential of MA-RRL in modeling coupled behaviors of the lane changer and follower.

Furthermore, the Transformer model (Vaswani et al., 2017) has been increasingly used in LC-related and driving behavior modeling studies in recent years (Gao et al., 2023; Guo et al., 2022; Li et al., 2022a; Zhu et al., 2022), which achieved better accuracy and efficiency with the attention mechanism and parallelized structure. It has also been introduced to RL and exceeded RRL on challenging memory environments in terms of interpretability and efficiency (Parisotto et al., 2020; Li et al., 2022a; Wei et al., 2022b; Chen et al., 2023). Thus the Transformer network is adopted in this study to process the observation-action history and make the agent aware of the traffic environment. Additionally, most of the RL-based LC models were developed based on traffic simulation, as it is a noise-free interactive environment for the RL agent. Although the models achieved good performance in the simulation environment, they might not be able to reflect the real situations. It is critical to develop the model using large-scale real-world CV data.

3. Methodology

The LC process is formulated as a two-player Markov game in this study, which is similar to the assumption in game theory-based LC models. The lane changer decides to merge or wait, while the follower decides to give way or not. One player needs to choose an appropriate action according to the current situation and the other's action. Their actions change the situation subsequently, which will affect themselves in turn. Besides, the LC scenario has elements of both cooperation and competition. For example, each driver wants to complete the driving task (e.g. changing to the target lane or maintaining the desired speed), and all drivers try to avoid the collision. These mixed cooperative–competitive characteristics are expected to be captured by the MA-RL algorithm, which can model the coupled driving behaviors of the lane changer and follower.

MA-RL is a sub-field of reinforcement learning. It focuses on studying the behavior of multiple agents interacting in a shared environment, which is closely related to the game theory (Busoniu et al., 2008; Buşoniu et al., 2010). Compared to the traditional single-agent RL, teaching the agent to maximize the cumulative reward, the multiple agents would cooperate and/or compete, and learn to accomplish a particular task collectively. Besides, the single-agent RL might be poorly suited to the multi-agent system, as the agents' changing policies during training make the environment non-stationary for any individual agent.

The multi-agent deep deterministic policy gradient (MA-DDPG) (Lowe et al., 2017) is used to model the driving behaviors of the lane changer and follower in this study. It is based on the DDPG (Lillicrap et al., 2015), which is a model-free off-policy RL algorithm using deep function approximators that can learn policies in continuous action spaces. The framework of centralized training with decentralized execution is adopted in MA-DDPG to solve the multi-agent problem, as shown in Fig. 1. Specifically, the actor μ_i only has access to its local observation O_i. While the critic Q_i can use extra information (e.g. other agents' actions A_i) to facilitate training, which enables inferring other agents' policies from observations. This assumption is more flexible and realistic

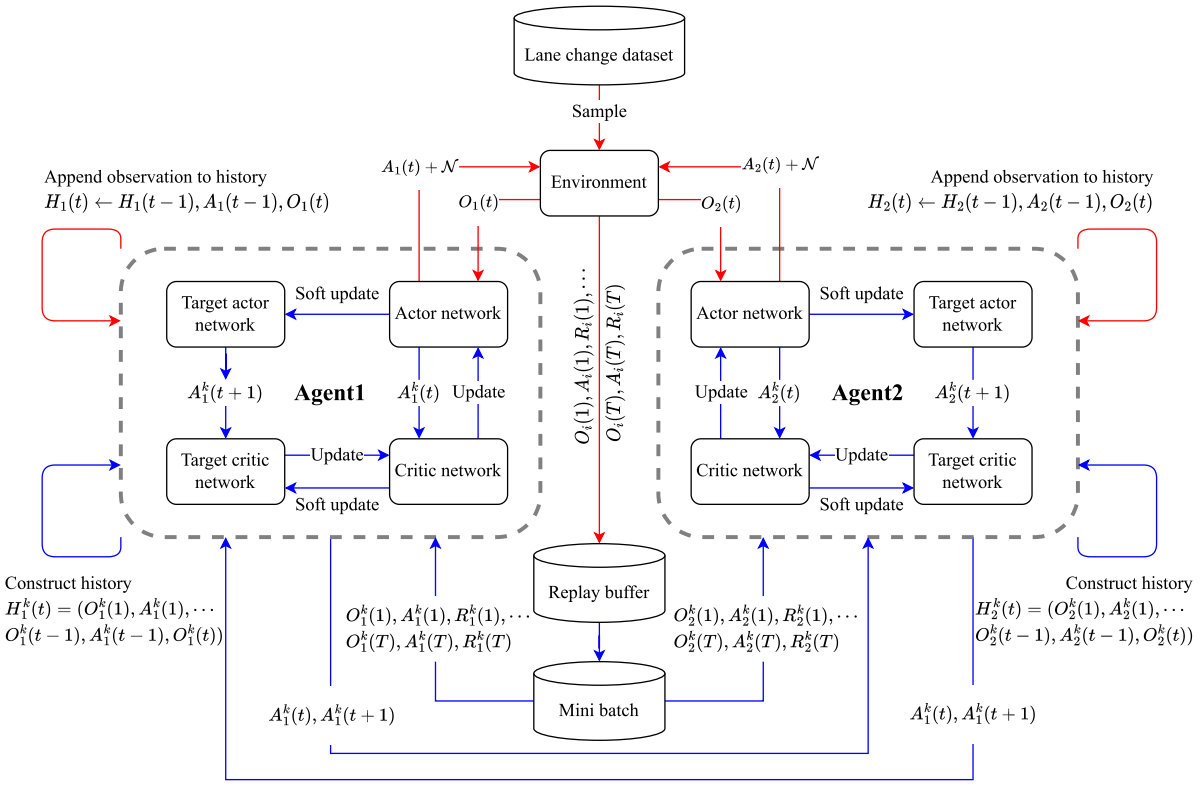


Fig. 2. Architecture of MA-TDDPG, including experience generation (red arrows) and training (blue arrows) procedure.

compared to knowing other agents' policies completely. The primary motivation behind this is that, if the actions taken by all agents are known, the environment is stationary even as the policies change. Therefore, the MA-DDPG method has the potential to be used for modeling the mixed cooperative–competitive driving behaviors of the lane changer and follower.

At the same time, it is expected that all driving behaviors are affected by memory, due to the existence of reaction time and vehicle inertia (Zhang, 2003; Yu and Shi, 2015; Pang et al., 2020). However, the traditional RL algorithms only utilize the single-step state without memory. The decision or action is made based on the current state only, disregarding the previous states. This partially observed single-step state may not contain enough information for the agent. It is advisable to add memory to the RL algorithms to achieve higher agent status and environment description accuracy. By implementing recurrent neural networks, the RRL is proposed to learn from the memory based on interactions with the environment (Moody and Wu, 1997). RRL has a more stable performance compared to RL when exposed to noisy datasets (Wang et al., 2020; Moody and Saffell, 2001). It is also more flexible in choosing the objective function and efficient in model training (Wang et al., 2018; Aboussalah and Lee, 2020). Thus, it is worth further exploring the potential of RRL in driving behavior modeling. Referring to the recurrent DPG (Heess et al., 2015) algorithms, we integrated the Transformer network into the framework of MA-DDPG and proposed MA-TDDPG, which would be used to model the lane changer's and follower's driving behaviors. The architecture of MA-TDDPG consisting of two agents is illustrated in Fig. 2, where the red arrows represent the experience generation processes, and the blue arrows represent the training processes.

In the MA-TDDPG model, each agent i consists of four Transformer networks, namely the actor (μ_i), target actor (μ_i'), critic (Q_i), and target critic (Q_i') network. The actor networks (μ and μ') interact with the environment according to the given state and their weights. Then, the critic networks (Q and Q') evaluate the action and output dummy Q-values, which would be used for the model update. The target networks (μ' and Q') are the lagged versions of the actual agent networks (μ and Q), which are used to improve the algorithm's stability with a soft update.

The LC scenario of lane changer and follower is considered as a partially observed Markov decision process (POMDP) of N agents, because: (i) the agent could only receive its private observation rather than the entire state; and (ii) the agent could only observe the underlying state indirectly through its observation-action history. The basic scenario of this Markov game is presented in Fig. 3.

The leader is defined as agent0, which participates in the game indirectly as a component of the environment. It would follow a fixed trajectory extracted from the data. The lane changer is agent1 and the follower is agent2. They play the game and interact with each other. The CV records the whole LC process as the follower. The longitudinal and lateral components of distance (d) and speed (v) are denoted as d^y, d^x and v^y, v^x, respectively. The main notations used in this study are summarized in Table 1.

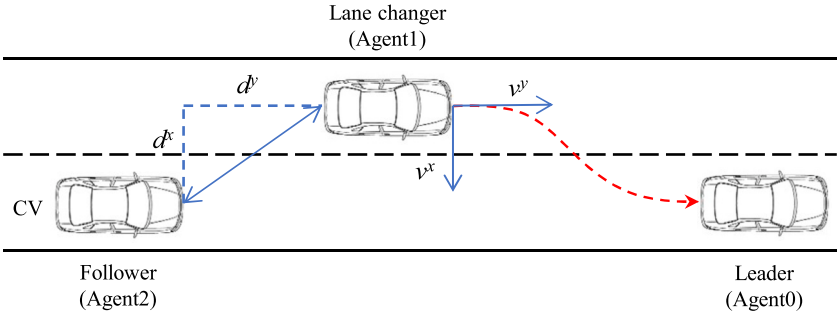


Fig. 3. Basic lane change scenario.

Table 1
Summary of notations.

Symbol	Definition
i, j	Vehicle and agent index (0 — leader; 1 — lane changer; 2 — follower)
t, T	Time
Δt	Time interval
$v_i^y(t)$	Longitudinal speed of vehicle i at time t
$v_i^x(t)$	Lateral speed of vehicle i at time t
$d_{ij}^y(t)$	Longitudinal distance between vehicle i and j at time t
$d_{ij}^x(t)$	Lateral distance between vehicle i and j at time t
$\Delta v_{ij}^y(t)$	Longitudinal relative speed of vehicle i and j at time t
$\Delta v_{ij}^x(t)$	Lateral relative speed of vehicle i and j at time t
$a_i^y(t)$	Longitudinal acceleration of vehicle i at time t
$a_i^x(t)$	Lateral acceleration of vehicle i at time t
$O_i(t)$	Observation of agent i at time t
$A_i(t)$	Action of agent i at time t
$H_i(t)$	Observation-action history of agent i at time t
$R_i(t)$	Reward of agent i at time t

At each time step t, each agent i would receive its local observation $O_i(t)$ consisting of the kinematics of the ego vehicle, and the distance and relative speed to the other two vehicles. For the lane changer (agent1), the observation at time t is given as

$$O_1(t) = (v_1^y(t), v_1^x(t), d_{01}^y(t), d_{01}^x(t), \Delta v_{01}^y(t), \Delta v_{01}^x(t), d_{12}^y(t), d_{12}^x(t), \Delta v_{12}^y(t), \Delta v_{12}^x(t)) \tag{1}$$

where $v_1^y(t)$ and $v_1^x(t)$ are the longitudinal and lateral speed of agent1, respectively. $d_{01}^y(t)$, $d_{01}^x(t)$, $\Delta v_{01}^y(t)$, and $\Delta v_{01}^x(t)$ are the longitudinal distance, lateral distance, longitudinal relative speed, and lateral relative speed between agent0 and agent1, respectively. Similarly, $d_{12}^y(t)$, $d_{12}^x(t)$, $\Delta v_{12}^y(t)$, and $\Delta v_{12}^x(t)$ describe the distance and relative speed between agent1 and agent2. The observation of the follower (agent2) is

$$O_2(t) = (v_2^y(t), v_2^x(t), d_{02}^y(t), d_{02}^x(t), \Delta v_{02}^y(t), \Delta v_{02}^x(t), d_{12}^y(t), d_{12}^x(t), \Delta v_{12}^y(t), \Delta v_{12}^x(t)) \tag{2}$$

For model simplification without losing generalizability, we assumed that the lane changer can move both longitudinally and laterally, while the follower can move only longitudinally. This assumption about the follower's motion has been adopted in many related studies based on CF theory (Laval and Leclercq, 2008; Duret et al., 2011; Zheng et al., 2013). Besides, the 95% quantile of the follower's lateral speed is 0.165 m/s in the extracted data, which makes the assumption acceptable. The lane changer's action is $A_1(t) = (\hat{a}_1^y(t), \hat{a}_1^x(t))$, and the follower's action is $A_2(t) = \hat{a}_2^y(t)$, where $\hat{a}^y(t)$ and $\hat{a}^x(t)$ are the longitudinal and lateral acceleration, respectively.

After receive the observation $O_i(t)$, the agent would summarize the history $H_i(t) = (O_i(1), A_i(1), \ldots, O_i(t-1), A_i(t-1), O_i(t))$ by combining the memory and current observation. Then, each agent selects an action $A_i(t) = \mu_i(H_i(t)|\theta_i^\mu)$ based on the history and policy. A random noise $\mathcal{N}$, generated by the Ornstein–Uhlenbeck process ($\theta = 0.15$ and $\sigma = 0.2$) (Uhlenbeck and Ornstein, 1930), would be added to the action for exploration. This temporally correlated noise could achieve a better exploration of physical environments with momentum. Then, all the actions are executed and the whole environment would be updated based on the

kinematic model, which is given by

$$
\begin{aligned}
&\hat{v}_0^y(t+1) = v_0^y(t)\\
&\hat{v}_0^x(t+1) = v_0^x(t)\\
&\hat{v}_i^y(t+1) = v_i^y(t) + \hat{a}_i^y(t)\Delta t\Big|_{i=1,2}\\
&\hat{v}_i^x(t+1) = v_i^x(t) + \hat{a}_i^x(t)\Delta t\Big|_{i=1,2}\\
&\Delta\hat{v}_{ij}^y(t+1) = \hat{v}_i^y(t+1) - \hat{v}_j^y(t+1)\Big|_{i=0,1;j=1,2;i<j}\\
&\Delta\hat{v}_{ij}^x(t+1) = \hat{v}_i^x(t+1) - \hat{v}_j^x(t+1)\Big|_{i=0,1;j=1,2;i<j}\\
&\hat{d}_{ij}^y(t+1) = d_{ij}^y(t) + \frac{v_i^y(t)+\hat{v}_i^y(t+1)}{2}\Delta t - \frac{v_j^y(t)+\hat{v}_j^y(t+1)}{2}\Delta t\Big|_{i=0,1;j=1,2;i<j}\\
&\hat{d}_{ij}^x(t+1) = d_{ij}^x(t) + \frac{v_i^x(t)+\hat{v}_i^x(t+1)}{2}\Delta t - \frac{v_j^x(t)+\hat{v}_j^x(t+1)}{2}\Delta t\Big|_{i=0,1;j=1,2;i<j}
\end{aligned}
\tag{3}
$$

where the hatted values represent the estimated values and the non-hatted values are observed ones. Δt is the update time interval, which is 0.1s in this study. The leader (agent0) is assumed to maintain its speed as it does not participate in the Markov game directly. Two reward functions are designed to make the agents imitate human drivers' behaviors based on speed and distance. The speed reward function is proposed to make the agents adopt proper actions to keep the same speed as a human driver, which is given by

$$
\begin{aligned}
R_{v1} &= -\Big(\frac{\hat{v}_1^y(t+1)-\mu_{v_1^y}}{\sigma_{v_1^y}} - \frac{v_1^y(t+1)-\mu_{v_1^y}}{\sigma_{v_1^y}}\Big)^2 - \Big(\frac{\hat{v}_1^x(t+1)-\mu_{v_1^x}}{\sigma_{v_1^x}} - \frac{v_1^x(t+1)-\mu_{v_1^x}}{\sigma_{v_1^x}}\Big)^2\\
R_{v2} &= -\Big(\frac{\hat{v}_2^y(t+1)-\mu_{v_2^y}}{\sigma_{v_2^y}} - \frac{v_2^y(t+1)-\mu_{v_2^y}}{\sigma_{v_2^y}}\Big)^2
\end{aligned}
\tag{4}
$$

where R_{v1} is the lane changer's speed reward and R_{v2} is the follower's. $\mu_{v_1^y}$, $\sigma_{v_1^y}$, $\mu_{v_1^x}$, $\sigma_{v_1^x}$, $\mu_{v_2^y}$, and $\sigma_{v_2^y}$ are the mean and standard deviation of v_1^y, v_1^x, and v_2^y, respectively. The normalized rewards are adopted to achieve a balance between multiple components, as well as multiple agents.

The distance reward function is used to let the agents maintain proper distances to other vehicles like a human driver, which is given by

$$
\begin{aligned}
R_{d1} = &-\Big(\frac{\hat{d}_{01}^y(t+1)-\mu_{d_{01}^y}}{\sigma_{d_{01}^y}} - \frac{d_{01}^y(t+1)-\mu_{d_{01}^y}}{\sigma_{d_{01}^y}}\Big)^2 - \Big(\frac{\hat{d}_{01}^x(t+1)-\mu_{d_{01}^x}}{\sigma_{d_{01}^x}} - \frac{d_{01}^x(t+1)-\mu_{d_{01}^x}}{\sigma_{d_{01}^x}}\Big)^2\\
&-\Big(\frac{\hat{d}_{12}^x(t+1)-\mu_{d_{12}^x}}{\sigma_{d_{12}^x}} - \frac{d_{12}^x(t+1)-\mu_{d_{12}^x}}{\sigma_{d_{12}^x}}\Big)^2\\
R_{d2} = &-\Big(\frac{\hat{d}_{12}^y(t+1)-\mu_{d_{12}^y}}{\sigma_{d_{12}^y}} - \frac{d_{12}^y(t+1)-\mu_{d_{12}^y}}{\sigma_{d_{12}^y}}\Big)^2
\end{aligned}
\tag{5}
$$

where R_{d1} and R_{d2} are the distance rewards of the lane changer and follower, respectively. R_{d1} consists of the longitudinal distance components (d_{01}^y and d_{12}^y) and lateral distance component (d_{12}^x). It should be noted that only the lateral distance between the lane changer and follower (d_{12}^x) is calculated. The reward function containing both d_{12}^x and d_{01}^x was tested. These two reward components were found to be highly correlated as the leader and follower are in the same lane. So d_{01}^x was removed for simplification and the model performance is not affected. Besides, only the longitudinal distance between the lane changer and follower (d_{12}^y) is considered in R_{d2}. We tried the reward function including d_{12}^y and d_{02}^y and found that the current one gave better results. This finding is consistent with the evidence that the follower's motions mainly depend on the lane changer (Zheng et al., 2013). In the end, the sequence $(O_i(1), A_i(1), R_i(1), \dots, O_i(T), A_i(T), R_i(T))$, consisting of observations, actions, and rewards of all agents, would be stored in the replay buffer.

At the beginning of the model training process, a random mini-batch of M observation-action trajectories $(O_i^k(1), A_i^k(1), R_i^k(1), \dots, O_i^k(T), A_i^k(T), R_i^k(T))$ are sampled for each agent. The history $H_i^k(t)$ is constructed by appending the current observation to the memory. Then, the target actor network μ_i' would choose an action $A_i^k(t+1) = \mu_i'(H_i^k(t+1))$ based on the policy. The target critic evaluates the policy of the target actor and outputs a scalar Q-value $Q_i'(H_i^k(t+1), A_1^k(t+1), \dots, A_N^k(t+1))$. It should be noted that the actions taken by all agents are used for one agent's policy evaluation, which keeps the environment stationary with changing policies. Similarly, the critic evaluates the actor's policy with $Q_i(H_i^k(t), A_1^k(t), \dots, A_N^k(t))$. Now we can update the critic network by minimizing the loss given by

$$
\mathcal{L}(\theta_i^Q) = \frac{1}{MT}\sum_k\sum_t (y_i^k(t) - Q_i(H_i^k(t), A_1^k(t), \dots, A_N^k(t)))^2 \tag{6}
$$

where

$$
y_i^k(t) = R_i^k(t) + \gamma Q_i'(H_i^k(t+1), A_1^k(t+1), \dots, A_N^k(t+1))|_{A_i^k(t+1)=\mu_i'(H_i^k(t+1))} \tag{7}
$$

The gradient used to update the actor network can be derived as

$$\nabla_{\theta_i^\mu} J \approx \frac{1}{MT} \sum_k \sum_t \nabla_{\theta_i^\mu} \mu_i(H_i^k(t)) \nabla_{A_i^k(t)} Q_i(H_i^k(t), A_1^k(t), \ldots, A_N^k(t))|_{A_i^k(t)=\mu_i(H_i^k(t))} \tag{8}$$

At last, the target actor and critic networks are soft updated with

$$\begin{aligned} \theta_i^{Q'} &\leftarrow \tau\theta_i^Q + (1-\tau)\theta_i^{Q'} \\ \theta_i^{\mu'} &\leftarrow \tau\theta_i^\mu + (1-\tau)\theta_i^{\mu'} \end{aligned} \tag{9}$$

The detailed MA-TDDPG algorithm is presented in Algorithm 1.

Algorithm 1 Multi-agent deep deterministic policy gradient for N agents (based on (Lillicrap et al., 2015; Heess et al., 2015; Lowe et al., 2017))

Randomly initialize recurrent critic network $Q_i(H_i(t), A_i(t)|\theta_i^Q)$ and recurrent actor network $\mu_i(H_i(t)|\theta_i^\mu)$ with weights θ_i^Q and θ_i^μ for each agent i
Initialize target networks Q_i' and μ_i' with weights $\theta_i^{Q'} \leftarrow \theta_i^Q$, $\theta_i^{\mu'} \leftarrow \theta_i^\mu$
Initialize replay buffer B
for episode =1 to M **do**
  Initialize history $H_i(0)$ for each agent i
  Initialize a random process $\mathcal{N}$ for action exploration
  for agent $i = 1$ to N **do**
    for $t = 1$ to T **do**
      Receive current observation $O_i(t)$
      $H_i(t) \leftarrow H_i(t-1), A_i(t-1), O_i(t)$ (append current observation and previous action to history)
      Select action $A_i(t) = \mu_i(H_i(t)|\theta_i^\mu) + \mathcal{N}$ according to the current policy and exploration noise
      Execute actions $\boldsymbol{A}(t) = (A_1(t), \cdots, A_N(t))$ and observe reward $R_i(t)$ and new observation $O_i(t+1)$
    end for
    Store the sequence $(O_i(1), A_i(1), R_i(1), \cdots, O_i(T), A_i(T), R_i(T))$ in replay buffer B
    Sample a random minibatch of M samples $(O_i^k(1), A_i^k(1), R_i^k(1), \cdots, O_i^k(T), A_i^k(T), R_i^k(T))$ from B
    Construct histories $H_i^k(t) = (O_i^k(1), A_i^k(1), \cdots, O_i^k(t-1), A_i^k(t-1), O_i^k(t))$
    Set $y_i^k(t) = R_i^k(t) + \gamma Q_i'(H_i^k(t+1), A_1^k(t+1), \cdots, A_N^k(t+1))|_{A_i^k(t+1)=\mu_i'(H_i^k(t+1))}$
    Update critic by minimizing the loss $\mathcal{L}(\theta_i^Q) = \frac{1}{MT}\sum_k \sum_t (y_i^k(t) - Q_i(H_i^k(t), A_1^k(t), \cdots, A_N^k(t)))^2$
    Update actor using the sampled policy gradient:
$$\nabla_{\theta_i^\mu} J \approx \frac{1}{MT} \sum_k \sum_t \nabla_{\theta_i^\mu} \mu_i(H_i^k(t)) \nabla_{A_i^k(t)} Q_i(H_i^k(t), A_1^k(t), \cdots, A_N^k(t))|_{A_i^k(t)=\mu_i(H_i^k(t))}$$
    Update target network parameters for each agent i:
$$\begin{aligned} \theta_i^{Q'} &\leftarrow \tau\theta_i^Q + (1-\tau)\theta_i^{Q'} \\ \theta_i^{\mu'} &\leftarrow \tau\theta_i^\mu + (1-\tau)\theta_i^{\mu'} \end{aligned}$$
  end for
end for

4. Data preparation

SPMD is the world's largest CV test program, which aims to demonstrate CV technologies in the real-world environment (Huang et al., 2017). 2842 equipped vehicles participated in the program for over 2 years in Ann Arbor, Michigan (Bezzina and Sayer, 2014). The two-month sample data (October 2012 and April 2013) are now available to the public on the ITS Data Hub (https://www.its.dot.gov/data/). The SPMD environment includes eight datasets, namely Data Acquisition System 1 (DAS1), Data Acquisition System 2 (DAS2), Basic Safety Message (BSM), Roadside Equipment, Network, Weather, Schedule, and Road Work Activity (Hamilton and Allen, 2015).

The DAS1 dataset, collected by the University of Michigan Transportation Research Institute (UMTRI) in April 2013, is used in this study. A total of 7960 trips recorded by 98 sedans equipped with the DAS1 and the MobilEye sensor (Harding et al., 2014) were investigated. Within the DAS1 dataset, the *DataLane* file records the CVs' lateral positions relative to lane boundaries (*LaneDistanceLeft* and *LaneDistanceRight*) and the estimated lane marking measurement quality (*LaneQualityLeft* and *LaneQualityRight*). The *DataFrontTargets* file provides relative positions (*Range* and *Transversal*) and relative speed (*RangeRate*) of the leading vehicles. The *DataWsu* file contains the geospatial (*LatitudeWsu* and *LongitudeWsu*) and kinematic (*GpsSpeedWsu* and *AxWsu*) information of CVs. The detailed description of the fields is reported in Table 2. All the data elements are collected at a frequency of 10 Hz. The *DataLane* and *DataFrontTargets* files were recorded by the MobilEye sensor, and the *DataWsu* file was collected by the GPS unit and controller area network (CAN) bus via wireless safety unit (WSU). Python programming language (Van Rossum and Drake, 2009) with Apache Spark bigdata analytic engine (Zaharia et al., 2016) was used for data manipulation.

To begin with, the invalid records were filtered out from the dataset. The criterion "*LaneQualityLeft* > *0* and *LaneQualityRight* > *0*" was used to remove records with poor lane marking measurement quality. The data points without valid GPS and Can Bus

Table 2
Field description of CV data.

Origin file	Field name	Description
Common	Device	A unique, numeric ID assigned to each DAS
	Trip	Count of ignition cycles—each ignition cycle commences when the ignition is in the on position and ends when it is in the off position
	Time	Time in centiseconds since DAS started, which (generally) starts when the ignition is in the on position (centisecond)
DataLane	LaneDistanceLeft ($d_l(t)$)	Distance between the center line of the vehicle and the left boundary of the travel lane (m)
	LaneDistanceRight ($d_r(t)$)	Distance between the center line of the vehicle and the right boundary of the travel lane (m)
	LaneQualityLeft	Quality of the estimated boundary measure of the travel lane's left boundary (ranging from 0 "very bad" to 3 "very good")
	LaneQualityRight	Quality of the estimated boundary measure of the travel lane's left boundary (ranging from 0 "very bad" to 3 "very good")
DataFrontTargets	ObstacleId	ID of new obstacle, as assigned by the Mobileye sensor, and its value will be the last used free ID
	TargetType	Classification of an identified obstacle/target (0: car; 1: truck; 2: motorcycle; 3: pedestrian; 4: bicycle)
	Range ($d_{i2}^{y}(t)$)	Longitudinal position of an object, typically the closest object, relative to a reference point on the host vehicle, according to the Mobileye sensor (m)
	RangeRate ($\Delta v_{i2}^{y}(t)$)	Longitudinal velocity of an object, typically the closest object, relative to the host vehicle, according to the Mobileye sensor (m/s)
	Transversal ($d_{i2}^{x}(t)$)	The lateral position of the obstacle, as determined by the Mobileye sensor (m)
DataWsu	GpsValidWsu	Communicates whether a GPS data point is valid (1) or not (0)
	LatitudeWsu	Latitude from WSU receiver (deg)
	LongitudeWsu	Longitude from WSU receiver (deg)
	GpsSpeedWsu ($v_2^{y}(t)$)	Speed from WSU GPS receiver (m/s)
	ValidCanWsu	Vehicle CAN Bus message to WSU is valid (1) or not (0)
	AxWsu	Longitudinal acceleration from vehicle CAN Bus vis WSU (m/s^2)

messages were cleaned with the criterion "*GpsValidWsu = 1* and *ValidCanWsu = 1*". Besides, the criterion "*TargetType = 0*" was applied to ensure that the front vehicles are cars. The driving behaviors would be different when the types of leaders differ (Ossen and Hoogendoorn, 2011), and the cars take up to 90.9% of all the front targets in the dataset. Therefore this study focuses on situations where the leaders are cars. After the cleaning process, the three datasets were merged based on the common fields*Device*, *Trip*, and *Time*.

Then, some records were filtered based on their values. According to the technical report (Kelly Blue Book, 2013), the fastest sedan can travel at a speed of 200 mph (90 m/s), and the maximum acceleration is around 7 m/s^2 in 2013. The records with *GpsSpeedWsu* value higher than 90 m/s, or *AxWsu* value greater than 7 m/s^2 were removed as outliers. The MobilEye sensor could cover three or more lanes and track multiple targets, including vehicles in the opposite direction. So the speed of the leader, calculated by *GpsSpeedWsu* + *Rangerate*, needs to be examined. The criterion "*GpsSpeedWsu* + *Rangerate* > *−1* m/s" was employed to filter out the records describing vehicles in the opposite direction. Besides, the free-flow scenarios were eliminated with the rule "*Range < 100* m". The vehicle pair with a lateral distance less than 2 m (*−2* m < *Transversal < 2* m) was assumed to be in the same lane and selected. If there is more than one car detected in the same lane, the vehicle with minimum longitudinal distance ($Range = min(Range)$) was determined to be the leader of CV.

Next, the data would be filtered to reduce the influence of measurement error and noise. The measurement quality of lane distance data ($LaneDistanceLeft$ and $LaneDistanceRight$) is not satisfactory, and the estimated quality ($LaneQualityLeft$ and $LaneQualityRight$) is usually less than 3 (Guo et al., 2022). Because the lane markings might be uncontinuous and unclear, and they could be shaded by other vehicles. A Gaussian filter was adopted to process the lane distance data for noise reduction. The raw and processed data are shown in Fig. 4. The computation of lateral speed is based on the lane distance data, which is given by Eq. (10). It is obvious that the Gaussian filter can effectively reduce the fluctuation of lateral speed without changing the lateral distance significantly.

Now the dataset contains information about the closest leader of CV. If the leader change (identified by the change of *ObstacleId*) and the distance decrease ($Range(t + \Delta t) < Range(t)$), there would be two possible scenarios: (i) CV changes lane; and (ii) another vehicle changes lane and appears in front of CV. In the latter case, CV is the follower in LC. It could record the movements of all participants during the anticipation period of LC. So these data would be used to model the driving behavior of the lane changer and follower. Based on our previous LC detection results (Guo et al., 2021), the events where CV changes lane were excluded, and the useful data were kept.

Further, the impact of the lane changer on the follower lasts for 25 s on average during the whole LC process (Ma and Ahn, 2008; Wang and Coifman, 2008). Zheng et al. (2013) pointed out that the average duration for the anticipation and relaxation stages of LC is 8–14 s and 10–15 s, respectively. In the dataset, the lane changer would block the leader from the CV's view after it enters the target lane. There are not enough data to model the coupled driving behaviors in the relaxation phase. Thus this study focuses on the anticipation stage. The LC scenarios within 10 s before the change of leader were extracted.

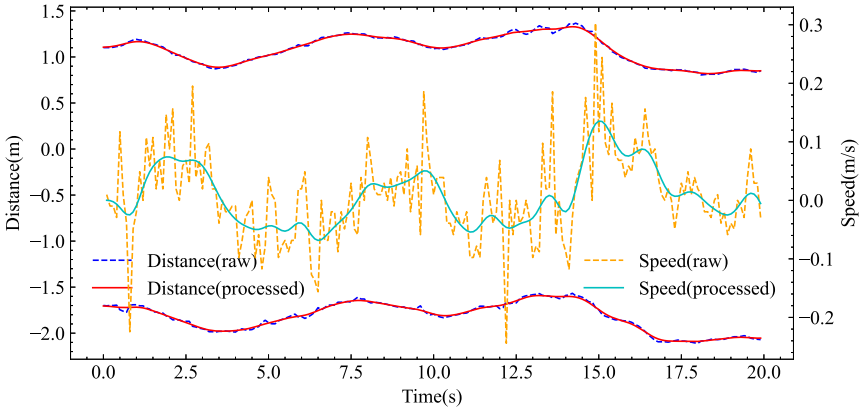


Fig. 4. The raw and processed lateral distance and speed data.

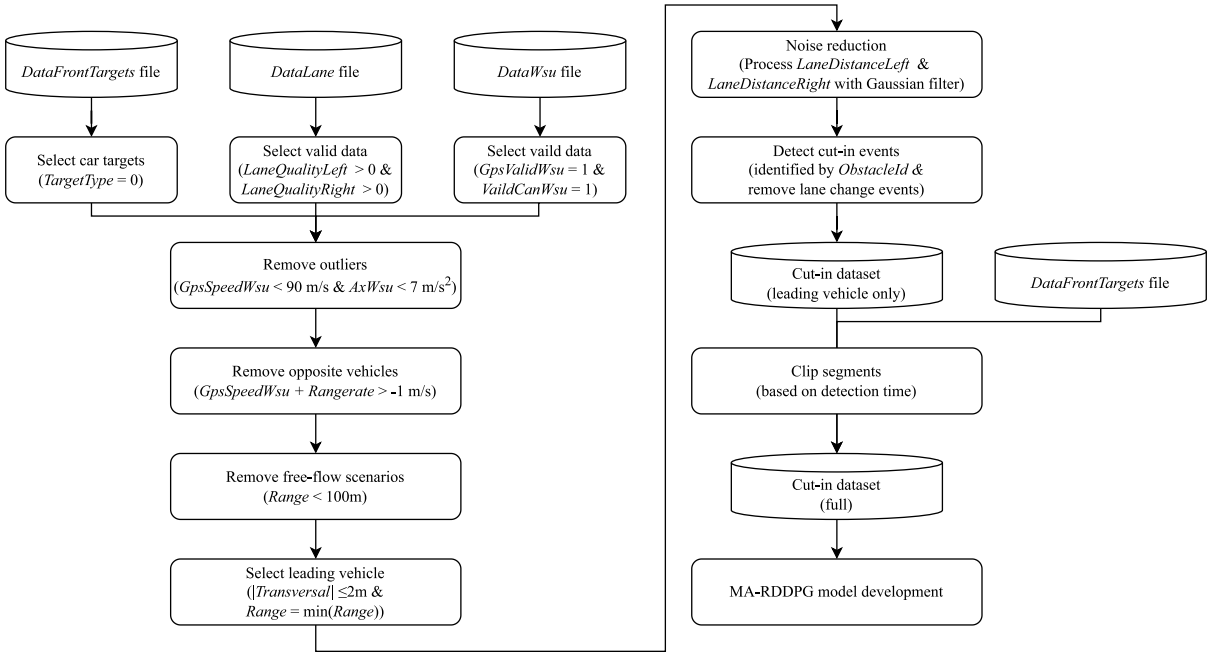


Fig. 5. Data preparation procedure.

At last, the LC event dataset was completed with the information of lane changers before LC. Now it only contains information about leaders, and lane changers were treated as new leaders after LC. By merging with the $DataFrontTargets$ dataset and selecting the records with $ObstacleId$ same as the leaders and lane changers, the LC scenario dataset can be created. It should be noted that the view angle of MobilEye's main camera is 28° (Stein et al., 2003). Some lane changers cannot be detected 10 s before LC, especially when the distances are small. In these situations, the segments were clipped to the time step when the lane changer is detected for the first time. After processing, there were 639 LC scenarios with an average length of 5.73 s extracted. 80% of the data would be used to train the MA-TDDPG lane changer and follower driving behaviors model, and the remaining 20% are kept for model testing. The detailed data preparation procedure is illustrated in Fig. 5.

In the processed LC dataset, the longitudinal speed of the follower $v_2^y(t) = GpsSpeedWsu$, the longitudinal distance $d_{i2}^y(t) = Range_{i2}\big|_{i=0,1}$, the lateral distance $d_{i2}^x(t) = Transversal_{i2}\big|_{i=0,1}$, and the longitudinal relative speed $\Delta v_{i2}^y(t) = Rangerate_{i2}\big|_{i=0,1}$ can be

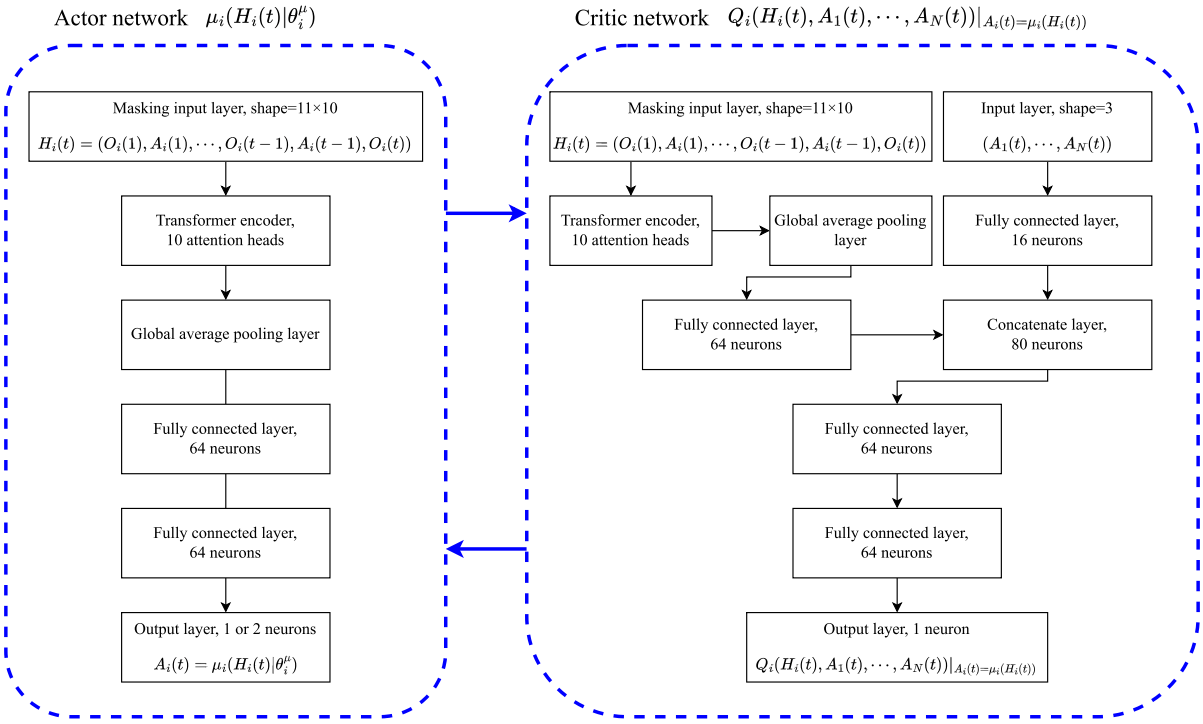


Fig. 6. Architecture of the actor and critic networks.

exported directly. Other variables used in the model are calculated as

$$
\begin{aligned}
v_2^x(t) &= \frac{(d_l(t) - d_l(t-\Delta t)) + (d_r(t) - d_r(t-\Delta t))}{2 \times \Delta t} \\
v_i^y(t) &= v_2^y(t) + \Delta v_{i2}^y(t)\Big|_{i=0,1} \\
v_i^x(t) &= v_2^x(t) + \frac{d_{i2}^x(t) - d_{i2}^x(t-\Delta t)}{\Delta t}\Big|_{i=0,1} \\
d_{01}^y(t) &= d_{02}^y(t) - d_{12}^y(t) \\
d_{01}^x(t) &= d_{02}^x(t) - d_{12}^x(t) \\
\Delta v_{01}^y(t) &= v_0^y(t) - v_1^y(t) \\
\Delta v_{ij}^x(t) &= v_i^x(t) - v_j^x(t)\Big|_{i=0,1;j=1,2;i<j}
\end{aligned}
\tag{10}
$$

where Δt is the time interval between two continuous records, which is 0.1 s in the dataset.

5. Results and discussion

5.1. MA-TDDPG model training and results

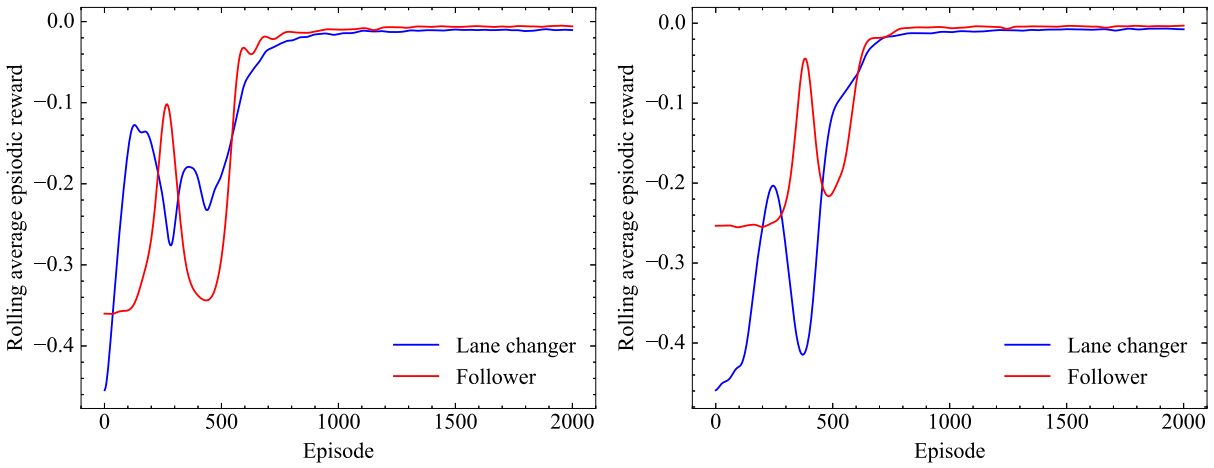
In this section, the MA-TDDPG model is trained and tested, whose structure is shown in Fig. 6. It is expected to learn the underlying policy of driver's behavior through the interaction with the environment and other agents. The memory window is assumed to be 1s and therefore the CV data collected in the past 10 time steps are used to develop the model. The input of the actor network consists of the observation-action history in the last 9 time steps and the observation at the current time step. The masking layer is adopted to mask the empty observation-action vectors and skip time steps at the beginning stage of training. It also masks the empty action at the current time step. Next, the Transformer encoder (Vaswani et al., 2017; Guo et al., 2022) is used to process the sequential history data and extract useful information, which has shown superior performance in time-series data processing. It should be noted that the size of the output layer is different for the lane changer and follower. Specifically, the longitudinal acceleration $A_2(t) = \hat{a}_2^y(t)$ is the only output of the follower (agent2), and the longitudinal and lateral acceleration $A_1(t) = (\hat{a}_1^y(t), \hat{a}_1^x(t))$ are two outputs of the lane changer (agent1).

The generated actions of all agents would be passed to the critic network as a part of the input. The other part is the same observation-action history as the actor network. Similarly, the Transformer encoder is used to deal with the observation-action

Table 3
Hyperparameters in MA-TDDPG model.

Hyperparameter	Value	Description
Actor learning rate	0.0005[a]	Learning rate used by the Adam optimizer of actor network
Critic learning rate	0.001[a]	Learning rate used by the Adam optimizer of critic network
Discount factor (γ)	0.9	Discount factor in DDPG
Soft target update rate (τ)	0.01	Soft update rate for target networks
Replay memory size	10,000	Number of training samples in replay memory
Batch size	256	Number of training samples used for gradient update

[a] Decay every 200 steps with a base of 0.95.



(a) MA-TDDPG with speed reward

(b) MA-TDDPG with distance reward

Fig. 7. Rolling average episodic reward of MA-TDDPG models during training.

history. Then these two inputs would be concentrated. The output of the critic network is a value evaluating the policy, which will be used for actor network updating. The target networks are the lag versions of actor and critic networks, which are used to make the model stable by updating softly. The hyperparameters adopted for the MA-TDDPG model training are presented in Table 3. The exponential decay learning rate schedule is adopted in this model to improve the training stability. The model is expected to explore the policy with a higher learning rate at the beginning stage, while find the best policy with the learned experience at a lower rate.

The MA-TDDPG model is trained using real-world data, in order to make the model imitate human driving behaviors. In each training episode, an LC scenario was randomly sampled from the training set and fed to the MA-TDDPG model. The model would process the input sequentially and store a set of experience $(O_i(1), A_i(1), R_i(1), \cdots, O_i(T), A_i(T), R_i(T))$ in the replay buffer. Next, a minibatch of samples was sampled and used for model training. After the network parameters update, another sample would be selected and passed to the model. The training process was repeated for 2000 episodes, where an episode means loading an LC event. The average episodic reward with a rolling window of 50 episodes during training is shown in Fig. 7.

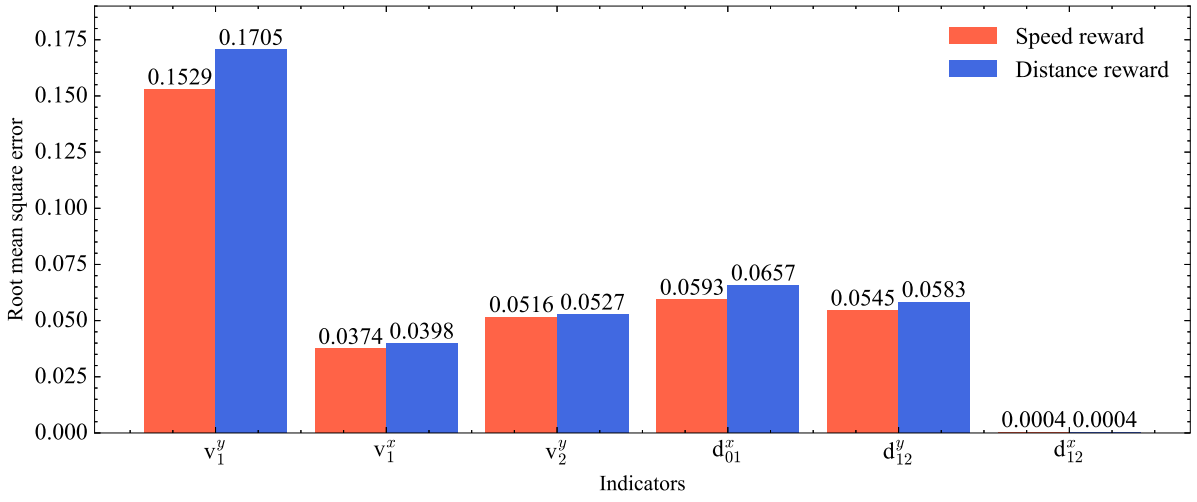
It is clear that the agents in MA-TDDPG models start to converge after around 800 episodes. The alternate change of the lane changer's and follower's rewards indicates the interaction during training. For example, the first peak of the red line in Fig. 7(a) could be a locally optimal point for the follower, which is not an equilibrium point for all agents. Thus the learned policy passes wrong information to the lane changer and makes its reward worse. At last, the follower reaches convergence first and leads the lane changer to converge with the correct policy.

The root mean square error (RMSE) and the Jensen–Shannon divergence (JSD) are employed to evaluate the performances of agents in both the longitudinal and lateral motions. The RMSE computes the average error between the simulated and observed values as given by Fig. 8(a).

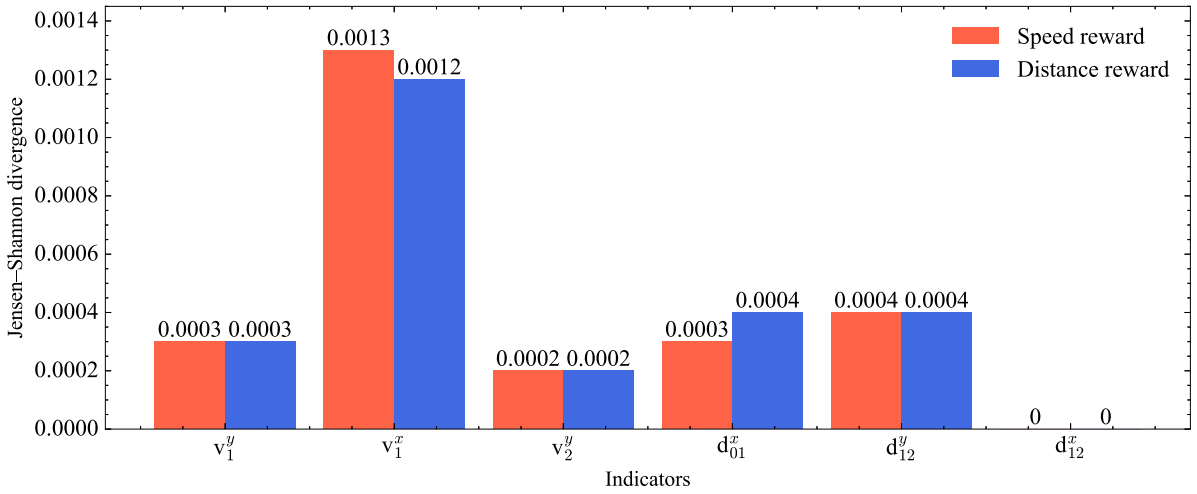
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2} \tag{11}$$

where $\hat{x}_i$ and x_i are the estimated and observed values, respectively. The RMSE of speed (v_1^y, v_1^x and v_2^y) and distance (d_{01}^y, d_{12}^y and d_{12}^x) are reported in Fig. 8(a). The Jensen–Shannon divergence (JSD) is used to measure the similarity between the simulated and observed value distributions, which is calculated as

$$JSD(P \parallel Q) = \frac{1}{2}\sum_{x\in\mathcal{X}} P(x)log(\frac{P(x)}{\frac{1}{2}(P(X)+Q(X))}) + \frac{1}{2}\sum_{x\in\mathcal{X}} Q(x)log(\frac{Q(x)}{\frac{1}{2}(P(X)+Q(X))}) \tag{12}$$



(a) Root mean square error (RMSE)



(b) Jensen–Shannon divergence (JSD)

Fig. 8. Root mean square error (RMSE) and Jensen–Shannon divergence (JSD) of MA-TDDPG models.

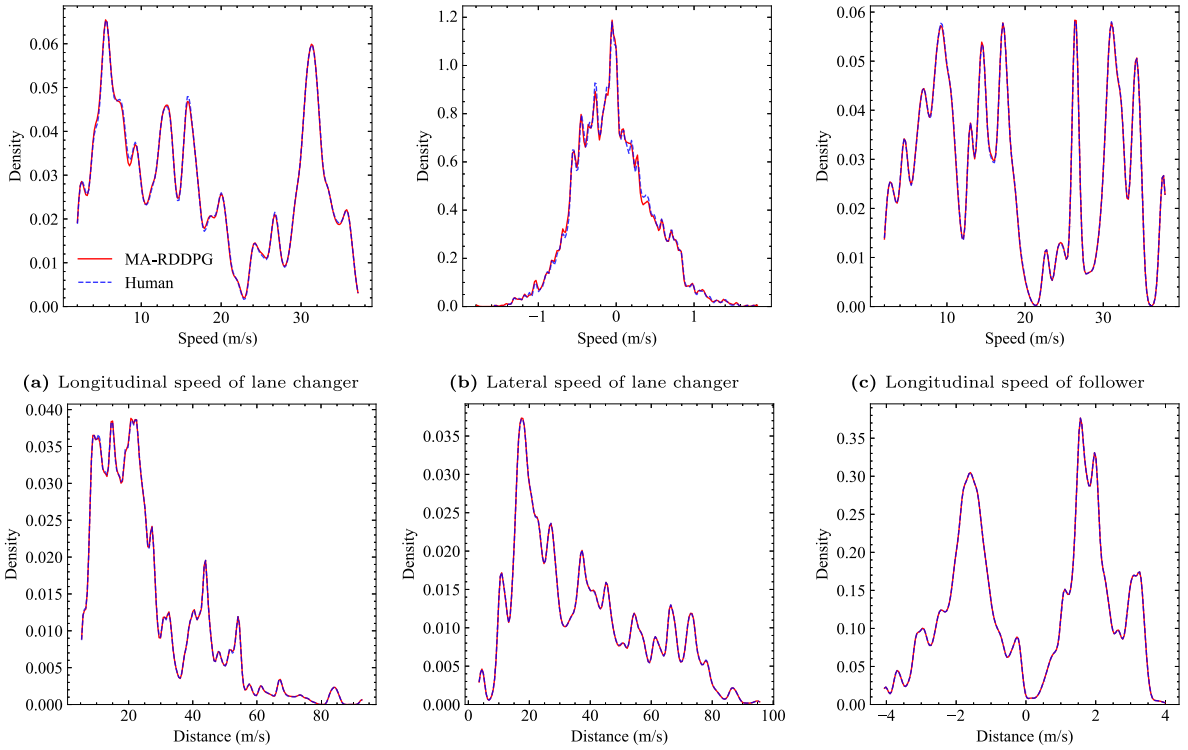
where P and Q are the estimated and actual discrete probability distributions, respectively. The JSD values are reported in Fig. 8(b).

In general, the MA-TDDPG models can achieve good performances in modeling the coupled driving behaviors of the lane changer and follower, indicated by their low JSD values of all the indicators. The speed reward outperforms the distance reward, as it has lower RMSE values, especially on the speed-related indicators. The distributions of the values estimated by the MA-TDDPG model with speed reward are compared with the real data, as shown in Fig. 9. The generated distributions are highly consistent with the observed human-driving data.

A possible reason for the performance difference is related to the state updating function, which is given by Eq. (3). The speed of the leader is assumed to be constant within one time step, which is not accurate in real situations. This error would be propagated to the model through the distance reward function, as the distances between the leader and lane changer are components. In contrast, the speed reward function depends on the speeds of the lane changer and follower, which is not influenced by this assumption. Thus the MA-TDDPG model achieves better performance with the speed reward function.

5.2. Trajectory reconstruction

To demonstrate the implementation of the proposed MA-TDDPG model, several general and corner LC scenarios were extracted and the trajectories of lane changer and follower were estimated. In the trajectory replication process, the leader would move along the recorded trajectory. The initial observations were passed to the lane changer and follower. Then the whole trajectories would be generated sequentially by the MA-TDDPG model. The observed and estimated two-dimensional, longitudinal, and lateral trajectories



(a) Longitudinal speed of lane changer (b) Lateral speed of lane changer (c) Longitudinal speed of follower

(d) Longitudinal distance between leader and lane changer (e) Longitudinal distance between lane changer and follower (f) Lateral distance between lane changer and follower

Fig. 9. Distributions of real and estimated values.

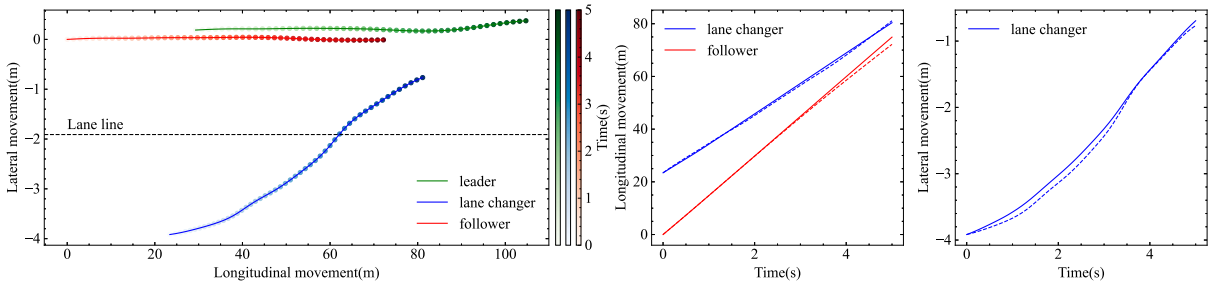
of the leader, lane changer, and follower are illustrated in Fig. 10. Specifically, Fig. 10(a) and (b) show the general left and right LC scenarios, respectively. Fig. 10(c) shows the LC case where the leading vehicle has the maximum speed in the dataset, Fig. 10(d) demonstrates the case with the highest following vehicle speed, and Fig. 10(e) presents the LC scenario with the minimum gap. The time gap between two continuous points in the two-dimensional trajectory is 0.1s.

It is clear that the trajectories estimated by the MA-TDDPG model are highly consistent with human-driven ones. The proposed model can accurately replicate the longitudinal and lateral accelerations of the lane changer and follower. Both the left and right LCs, as well as the general and corner cases, can be modeled in the unified model, which indicates its great generalizability and potential for practical use.

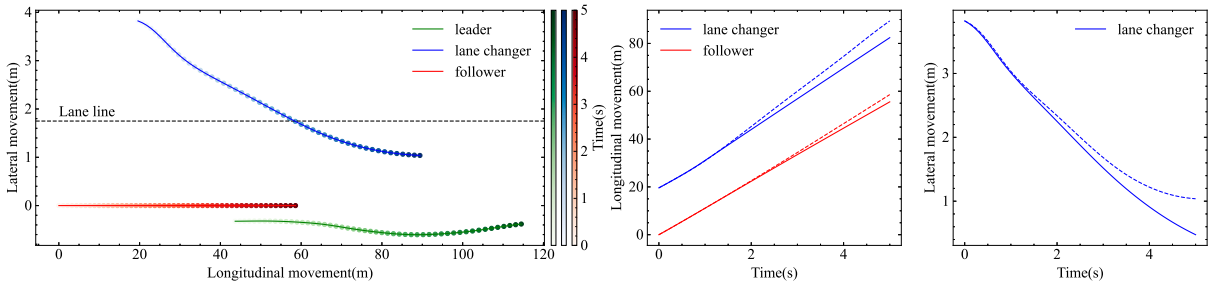
5.3. Ablation and comparison study

To further reveal the advantages of the MA-TDDPG model, the ablation and comparison study is carried out for investigation. Certain components are removed systematically to understand the contribution of the component to the overall performance. The baseline RL models include DDPG, MA-DDPG, TDDPG, and full-direction MA-TDDPG (MA-TDDPG(full)). Specifically, DDPG is the basic model built with a neural network (NN), which only uses the current observation for action choosing without considering the interaction between agents. MA-DDPG and TDDPG are two extensions of the DDPG model, in which the multiple agents' interactions and memory effects are considered, respectively. The MA-TDDPG (full) model includes both the longitudinal and lateral movements of the follower, while the follower is only allowed to move longitudinally in the MA-TDDPG model. The MA-TDDPG (full) model is constructed to analyze the influences of different vehicle action assumptions on model performances. The speed reward function is used for all these RL models. Two supervised learning models, namely Transformer and NN, are trained to investigate the contribution of the RL framework. Their network structures are the same as the intelligent agent in the corresponding RL models. These models would clone the longitudinal and lateral acceleration of the lane changer and the follower directly. Additionally, a supervised learning model (LSTM) and a multi-agent recurrent DDPG model (MA-RDDPG (LSTM)) based on the widely used LSTM network are trained for comparison. The performances of all models are reported in Table 4.

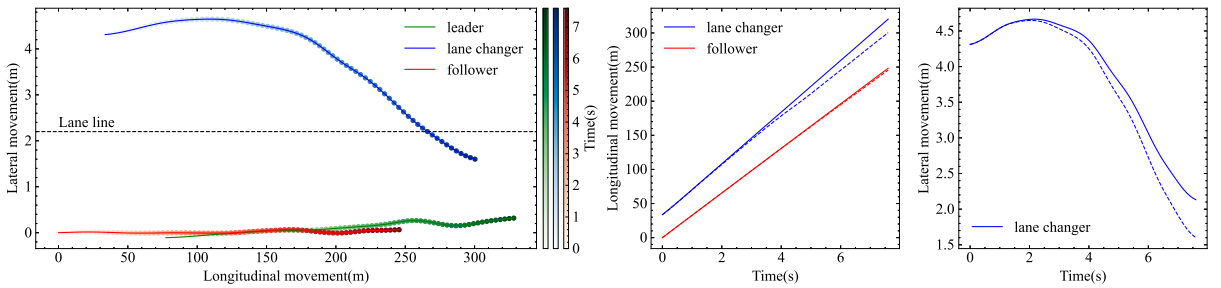
In the ablation study, the basic DDPG model cannot achieve a good performance, which indicates that the single-step observation and non-interactive environment do not contain enough information and exact assumptions for the modeling task. The MA-DDPG and TDDPG models outperform the DDPG model. The memory-capable structure seems to contribute more to the accuracy improvement,



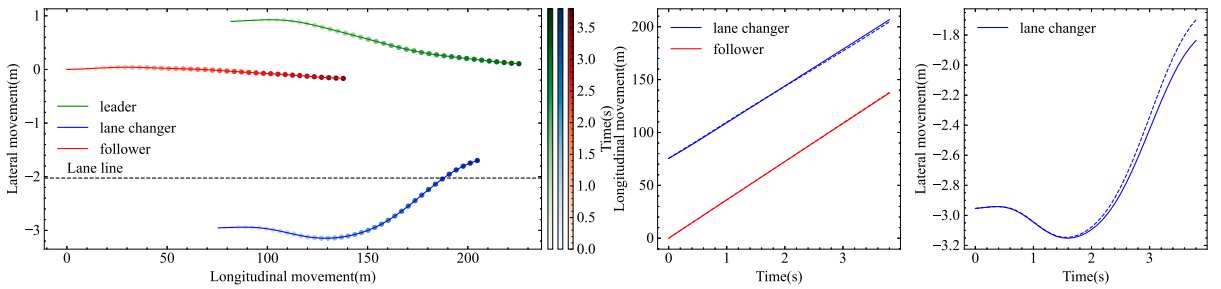
(a) Left lane change scenario



(b) Right lane change scenario



(c) Leading vehicle with maximum speed

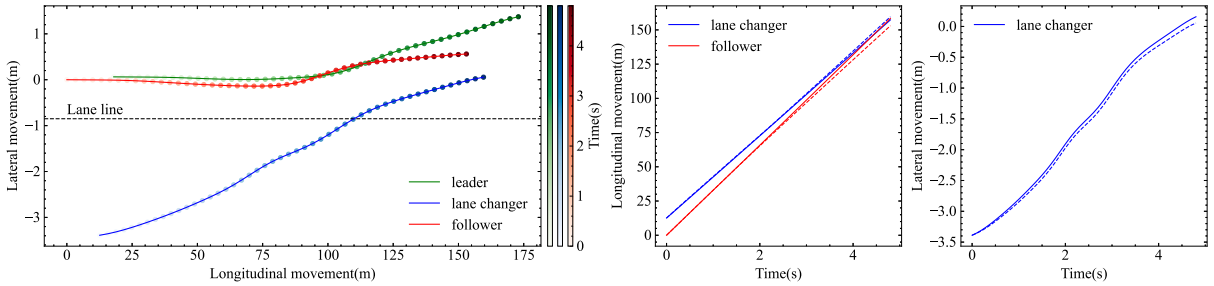


(d) Following vehicle with maximum speed

Fig. 10. Trajectory reconstruction examples (solid line — estimated trajectory, dashed line — observed trajectory).

according to the smaller RMSE values of TDDPG compared to MA-DDPG. The MA-TDDPG model is the combination of MA-DDPG and TDDPG models and reaches the best performance, which indicates the importance of multi-agent and memory-capable architectures. Compared to supervised learning models, the corresponding DDPG and TDDPG models achieve lower RMSE values than the NN and Transformer models, respectively. The RL structure can learn the underlying policy of driving behaviors rather than simple imitation, which makes them perform better when modeling unseen driving behaviors in the testing set. Besides, the Transformer-based models have better accuracy in both the supervised learning and MA-RRL framework. The computation time in action generation of the MA-TDDPG model (16 ms) is also less than the MA-RDDPG (LSTM) model (22 ms). The attention mechanism and parallelized structure of the Transformer network show advantages in processing time-series data and modeling driving behaviors.

It is worth mentioning that the MA-TDDPG (full) model performance is not as good as the MA-TDDPG model, although its basic assumption might be more realistic. The performance in the lateral direction (v_1^x and d_{12}^x) is even worse than the DDPG model. A



(e) Lane change scenario with minimum gap

Fig. 10. (*continued*).

Table 4
Root mean square error (RMSE) of models.

Indicator (RMSE)	v_1^y	v_1^x	v_2^y	d_{01}^y	d_{12}^y	d_{12}^x
MA-TDDPG	0.1529	0.0374	0.0516	0.0593	0.0545	0.0004
MA-RDDPG(LSTM)	0.1566	0.0396	0.0536	0.0625	0.0569	0.0004
MA-TDDPG(full)	0.1666	0.0607	0.0619	0.0650	0.0611	0.0037
MA-DDPG	0.1975	0.0477	0.0706	0.0654	0.0614	0.0007
TDDPG	0.1620	0.0397	0.0551	0.0626	0.0592	0.0006
DDPG	0.2013	0.0464	0.0967	0.0753	0.0648	0.0008
Transformer	0.1725	0.0424	0.0612	0.0627	0.0628	0.0014
LSTM	0.1806	0.0455	0.0689	0.0634	0.0656	0.0014
NN	0.2241	0.0507	0.1082	0.0856	0.0711	0.0015

potential reason could be that the lateral movement of the follower is unconscious driving behavior without a specific purpose. There might not be a policy to be inferred and imitated. If the model is forced to learn these lateral actions, the agent would be confused and generate unrealistic actions, as evidenced by the large error in the lateral direction (v_1^x and d_{12}^x). These erroneous lateral actions will also affect the agents in choosing longitudinal actions, resulting in an accuracy decrease in the longitudinal direction.

6. Conclusion

This study adds to the literature by proposing the MA-TDDPG model to imitate the coupled driving behavior during LC. The interactive behaviors of both the lane changer and the follower are considered jointly. The MA-TDDPG algorithm can accommodate multi-vehicle interactions and observation-action memories, whose assumptions are consistent with human driving nature. Large-scale naturalistic driving data collected by CVs in the SPMD program are used in this study. The results indicate that the proposed approach could realistically replicate the longitudinal and lateral actions of both the lane changer and the follower. The RL models can learn the underlying policy and achieve better performances than the supervised learning models. The MA-TDDPG model outperforms the baseline RL models in terms of driving behavior imitation by considering both the multi-vehicle interaction and the memory effect. The Transformer-based models show better accuracy and efficiency than the widely used LSTM-based models. In the implementation, the vehicle trajectories replicated by the MA-TDDPG model are highly consistent with the observed ones.

Considering there will be a long-term mixture of autonomous vehicles and human-driven vehicles in the future, it is critical to model the coupled LC behavior of human drivers. The proposed model can be used to imitate human-like coupled LC behaviors, which is essential in safety-aware microscopic traffic simulations. The generated LC motions are more consistent with human driving habits and can also be used as references in autonomous/assistant driving systems. Besides, the current study creates an interactive simulation environment, as the reactions of the LC and following vehicles can be reproduced. Optimizing the driving strategies of autonomous vehicles (e.g., explicitly considering safety factors) will be the future direction and the current study has laid a foundation for it.

For future studies, the generalizability of the proposed models will be further tested using datasets in different environments. Advanced LC control strategies can be developed based on the proposed model. The responses of the following vehicle reproduced by the MA-TDDPG model can be incorporated into LC control by replacing the assumed actions of the follower (Wang et al., 2015). It is also beneficial for the development of cooperative LC models (Lin et al., 2019a; Ladino and Wang, 2020). Moreover, the leading vehicle in the target lane can be included in the cooperative LC control. It would accelerate to create the acceptable gap for LC if it is feasible. This strategy is expected to eliminate the negative impact on traffic flow caused by LC. As the reactions of the lane changer and the follower can be simulated by the proposed model, this novel cooperative LC control strategy is accessible.

CRediT authorship contribution statement

Hongyu Guo: Writing – review & editing, Validation, Software, Methodology, Investigation, Conceptualization. **Mehdi Keyvan-Ekbatani:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization. **Kun Xie:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Conceptualization.

References

Aboussalah, A.M., Lee, C.-G., 2020. Continuous control with stacked deep dynamic recurrent reinforcement learning for portfolio optimization. Expert Syst. Appl. 140, 112891.

Ahn, S., Cassidy, M.J., 2007. Freeway traffic oscillations and vehicle lane-change maneuvers. In: Transportation and Traffic Theory 2007.

Alexiadis, V., Colyar, J., Halkias, J., Hranac, R., McHale, G., 2004. The next generation simulation program. Inst. Transp. Eng. ITE J. 74 (8), 22.

Ali, Y., Bliemer, M.C., Zheng, Z., Haque, M.M., 2020. Cooperate or not? Exploring drivers' interactions and response times to a lane-changing request in a connected environment. Transp. Res. C 120, 102816.

Ali, Y., Zheng, Z., Haque, M.M., Wang, M., 2019. A game theory-based approach for modelling mandatory lane-changing behaviour in a connected environment. Transp. Res. C 106, 220–242.

Alsaleh, R., Sayed, T., 2021. Markov-game modeling of cyclist-pedestrian interactions in shared spaces: A multi-agent adversarial inverse reinforcement learning approach. Transp. Res. C 128, 103191.

Aradi, S., 2020. Survey of deep reinforcement learning for motion planning of autonomous vehicles. IEEE Trans. Intell. Transp. Syst..

Aslani, M., Mesgari, M.S., Wiering, M., 2017. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. Transp. Res. C 85, 732–752.

Bevly, D., Cao, X., Gordon, M., Ozbilgin, G., Kari, D., Nelson, B., Woodruff, J., Barth, M., Murray, C., Kurt, A., et al., 2016. Lane change and merge maneuvers for connected and automated vehicles: A survey. IEEE Trans. Intell. Veh. 1 (1), 105–120.

Bezzina, D., Sayer, J., 2014. Safety Pilot Model Deployment: Test Conductor Team Report. Technical Report DOT HS 812 171, National Highway Traffic Safety Administration.

Busoniu, L., Babuska, R., De Schutter, B., 2008. A comprehensive survey of multiagent reinforcement learning. IEEE Trans. Syst., Man, Cybern. C (Appl. Rev.) 38 (2), 156–172.

Buşoniu, L., Babuška, R., Schutter, B.D., 2010. Multi-agent reinforcement learning: An overview. In: Innovations in Multi-Agent Systems and Applications-1. Springer, pp. 183–221.

Butakov, V.A., Ioannou, P., 2014. Personalized driver/vehicle lane change models for ADAS. IEEE Trans. Veh. Technol. 64 (10), 4422–4431.

Chen, Y., Dong, C., Palanisamy, P., Mudalige, P., Muelling, K., Dolan, J.M., 2019. Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

Chen, K., Liu, P., Li, Z., Wang, Y., Lu, Y., 2021. Modeling anticipation and relaxation of lane changing behavior using deep learning. Transp. Res. Rec. 2675 (12), 186–200.

Chen, L., Wang, Y., Miao, Z., Mo, Y., Feng, M., Zhou, Z., Wang, H., 2023. Transformer-based imitative reinforcement learning for multi-robot path planning. IEEE Trans. Ind. Inform..

Dong, J., Chen, S., Li, Y., Du, R., Steinfeld, A., Labi, S., 2021. Space-weighted information fusion using deep reinforcement learning: The context of tactical control of lane-changing autonomous vehicles and connectivity range assessment. Transp. Res. C 128, 103192.

Duret, A., Ahn, S., Buisson, C., 2011. Passing rates to measure relaxation and impact of lane-changing in congestion. Comput.-Aided Civ. Infrastruct. Eng. 26 (4), 285–297.

Essa, M., Sayed, T., 2020. Self-learning adaptive traffic signal control for real-time safety optimization. Accid. Anal. Prev. 146, 105713.

Farazi, N.P., Zou, B., Ahamed, T., Barua, L., 2021. Deep reinforcement learning in transportation research: A review. Transp. Res. Interdiscip. Perspect. 11, 100425.

Gao, K., Li, X., Chen, B., Hu, L., Liu, J., Du, R., Li, Y., 2023. Dual transformer based prediction for lane change intentions and trajectories in mixed traffic environment. IEEE Trans. Intell. Transp. Syst..

Gipps, P.G., 1986. A model for the structure of lane-changing decisions. Transp. Res. B 20 (5), 403–414.

Guo, H., Keyvan-Ekbatani, M., Xie, K., 2022. Lane change detection and prediction using real-world connected vehicle data. Transp. Res. C 142, 103785.

Guo, H., Xie, K., Keyvan-Ekbatani, M., 2021. Lane change detection using naturalistic driving data. In: 2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems. MT-ITS, IEEE, pp. 1–6.

Guo, H., Xie, K., Keyvan-Ekbatani, M., 2023. Modeling driver's evasive behavior during safety–critical lane changes: Two-dimensional time-to-collision and deep reinforcement learning. Accid. Anal. Prev. 186, 107063.

Hamilton, B., Allen, 2015. Safety Pilot Model Deployment-Sample Data Environment Data Handbook. Technical Report, US department of transportation.

Harding, J., Powell, G., Yoon, R., Fikentscher, J., Doyle, C., Sade, D., Lukuc, M., Simons, J., Wang, J., et al., 2014. Vehicle-To-Vehicle Communications: Readiness of V2V Technology for Application. Technical Report DOT HS 812 014, National Highway Traffic Safety Administration.

Haydari, A., Yilmaz, Y., 2020. Deep reinforcement learning for intelligent transportation systems: A survey. IEEE Trans. Intell. Transp. Syst..

He, X., Yang, H., Hu, Z., Lv, C., 2022. Robust lane change decision making for autonomous vehicles: An observation adversarial reinforcement learning approach. IEEE Trans. Intell. Veh. 8 (1), 184–193.

Heess, N., Hunt, J.J., Lillicrap, T.P., Silver, D., 2015. Memory-based control with recurrent neural networks. arXiv preprint arXiv:1512.04455.

Henclewood, D., Abramovich, M., Yelchuru, B., 2014. Safety Pilot Model Deployment–One Day Sample Data Environment Data Handbook. Technical Report, Research and Technology Innovation Administration, US Department of Transportation.

Hidas, P., 2002. Modelling lane changing and merging in microscopic traffic simulation. Transp. Res. C 10 (5–6), 351–371.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Huang, X., Zhao, D., Peng, H., 2017. Empirical study of DSRC performance based on safety pilot model deployment data. IEEE Trans. Intell. Transp. Syst. 18 (10), 2619–2628.

Jiang, S., Tran, C.Q., Keyvan-Ekbatani, M., 2024. Regional route guidance with realistic compliance patterns: Application of deep reinforcement learning and MPC. Transp. Res. C 158, 104440.

Jiang, L., Xie, Y., Evans, N.G., Wen, X., Li, T., Chen, D., 2022. Reinforcement Learning based cooperative longitudinal control for reducing traffic oscillations and improving platoon stability. Transp. Res. C 141, 103744.

Kelly Blue Book, 2013. Highest horsepower sedans of 2013. https://www.kbb.com/highest-horsepower-cars/sedan/2013/.

Kesting, A., Treiber, M., Helbing, D., 2007. General lane-changing model MOBIL for car-following models. Transp. Res. Rec. 1999 (1), 86–94.

Keyvan-Ekbatani, M., Knoop, V.L., Daamen, W., 2016a. Categorization of the lane change decision process on freeways. Transp. Res. C 69, 515–526.

Keyvan-Ekbatani, M., Knoop, V.L., Grébert, V., Daamen, W., 2016b. Lane change strategies on freeways: A microscopic simulation study. In: Traffic and Granular Flow'15. Springer, pp. 395–402.

Kita, H., 1999. A merging–giveway interaction model of cars in a merging section: a game theoretic analysis. Transp. Res. A 33 (3–4), 305–312.
Kumar, P., Perrollaz, M., Lefevre, S., Laugier, C., 2013. Learning-based approach for online lane change intention prediction. In: 2013 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 797–802.
Ladino, A., Wang, M., 2020. A dynamic game formulation for cooperative lane change strategies at highway merges. IFAC-PapersOnLine 53 (2), 15059–15064.
Laval, J.A., Daganzo, C.F., 2006. Lane-changing in traffic streams. Transp. Res. B 40 (3), 251–264.
Laval, J.A., Leclercq, L., 2008. Microscopic modeling of the relaxation phenomenon using a macroscopic lane-changing model. Transp. Res. B 42 (6), 511–522.
Leclercq, L., Chiabaut, N., Laval, J., Buisson, C., 2007. Relaxation phenomenon after lane changing: Experimental validation with NGSIM data set. Transp. Res. Rec. 1999 (1), 79–85.
Li, Y., 2017. Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274.
Li, G., Qiu, Y., Yang, Y., Li, Z., Li, S., Chu, W., Green, P., Li, S.E., 2022a. Lane change strategies for autonomous vehicles: a deep reinforcement learning approach based on transformer. IEEE Trans. Intell. Veh..
Li, G., Yang, Y., Li, S., Qu, X., Lyu, N., Li, S.E., 2022b. Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness. Transp. Res. C 134, 103452.
Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.
Lin, D., Li, L., Jabari, S.E., 2019a. Pay to change lanes: A cooperative lane-changing strategy for connected/automated driving. Transp. Res. C 105, 550–564.
Lin, Y., McPhee, J., Azad, N.L., 2019b. Longitudinal dynamic versus kinematic models for car-following control using deep reinforcement learning. In: 2019 IEEE Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 1504–1510.
Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I., 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. Adv. Neural Inf. Process. Syst. 30.
Ma, T., Ahn, S., 2008. Comparisons of speed-spacing relations under general car following versus lane changing. Transp. Res. Rec. 2088 (1), 138–147.
Moody, J., Saffell, M., 2001. Learning to trade via direct reinforcement. IEEE Trans. Neural Netw. 12 (4), 875–889.
Moody, J., Wu, L., 1997. Optimization of trading systems and portfolios. In: Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering. CIFEr, IEEE, pp. 300–307.
Newell, G.F., 2002. A simplified car-following theory: a lower order model. Transp. Res. B 36 (3), 195–205.
Ossen, S., Hoogendoorn, S.P., 2011. Heterogeneity in car-following behavior: Theory and empirics. Transp. Res. C 19 (2), 182–195.
Ozan, C., Baskan, O., Haldenbilen, S., Ceylan, H., 2015. A modified reinforcement learning algorithm for solving coordinated signalized networks. Transp. Res. C 54, 40–55.
Pang, M.-Y., Jia, B., Xie, D.-F., Li, X.-G., 2020. A probability lane-changing model considering memory effect and driver heterogeneity. Transp. B 8 (1), 72–89.
Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., Jaderberg, M., Kaufman, R.L., Clark, A., Noury, S., et al., 2020. Stabilizing transformers for reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 7487–7498.
Peake, A., McCalmon, J., Raiford, B., Liu, T., Alqahtani, S., 2020. Multi-agent reinforcement learning for cooperative adaptive cruise control. In: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence. ICTAI, IEEE, pp. 15–22.
Shi, K., Wu, Y., Shi, H., Zhou, Y., Ran, B., 2022. An integrated car-following and lane changing vehicle trajectory prediction algorithm based on a deep neural network. Phys. A 599, 127303.
Stein, G.P., Mano, O., Shashua, A., 2003. Vision-based ACC with a single camera: bounds on range and range rate accuracy. In: IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683). IEEE, pp. 120–125.
Sun, D., Kondyli, A., 2010. Modeling vehicle interactions during lane-changing behavior on arterial streets. Comput.-Aided Civ. Infrastruct. Eng. 25 (8), 557–571.
Toledo, T., 2007. Driving behaviour: models and challenges. Transp. Rev. 27 (1), 65–84.
Toledo, T., Koutsopoulos, H.N., Ben-Akiva, M.E., 2003. Modeling integrated lane-changing behavior. Transp. Res. Rec. 1857 (1), 30–38.
Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E 62 (2), 1805.
Uhlenbeck, G.E., Ornstein, L.S., 1930. On the theory of the Brownian motion. Phys. Rev. 36 (5), 823.
Van Rossum, G., Drake, F.L., 2009. Python 3 Reference Manual. CreateSpace, Scotts Valley, CA.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. arXiv preprint arXiv:1706.03762.
Wang, P., Chan, C.-Y., 2017. Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1–6.
Wang, C., Coifman, B., 2008. The effect of lane-change maneuvers on a simplified car-following theory. IEEE Trans. Intell. Transp. Syst. 9 (3), 523–535.
Wang, D., Fan, T., Han, T., Pan, J., 2020. A two-stage reinforcement learning approach for multi-UAV collision avoidance under imperfect sensing. IEEE Robot. Autom. Lett. 5 (2), 3098–3105.
Wang, M., Hoogendoorn, S.P., Daamen, W., van Arem, B., Happee, R., 2015. Game theoretic approach for predictive lane-changing and car-following control. Transp. Res. C 58, 73–92.
Wang, G., Hu, J., Li, Z., Li, L., 2021a. Harmonious lane changing via deep reinforcement learning. IEEE Trans. Intell. Transp. Syst..
Wang, P., Li, H., Chan, C.-Y., 2019b. Continuous control for automated lane change behavior based on deep deterministic policy gradient algorithm. In: 2019 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 1454–1460.
Wang, B., Li, Z., Wang, S., Li, M., Ji, A., 2022a. Modeling bounded rationality in discretionary lane change with the quantal response equilibrium of game theory. Transp. Res. B 164, 145–161.
Wang, Y., Wang, L., Guo, J., Papamichail, I., Papageorgiou, M., Wang, F.-Y., Bertini, R., Hua, W., Yang, Q., 2022b. Ego-efficient lane changes of connected and automated vehicles with impacts on traffic flow. Transp. Res. C 138, 103478.
Wang, L., Zhang, W., He, X., Zha, H., 2018. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2447–2456.
Wang, J., Zhang, Z., Lu, G., 2021b. A Bayesian inference based adaptive lane change prediction model. Transp. Res. C 132, 103363.
Wang, J., Zhang, Q., Zhao, D., Chen, Y., 2019a. Lane change decision-making through deep reinforcement learning with rule-based constraints. In: 2019 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–6.
Wei, X., Huang, X., Yang, L., Cao, G., Tao, Z., Wang, B., An, J., 2022b. Hierarchical RNNs-Based transformers MADDPG for mixed cooperative-competitive environments. J. Intell. Fuzzy Systems 43 (1), 1011–1022.
Wei, C., Hui, F., Yang, Z., Jia, S., Khattak, A.J., 2022a. Fine-grained highway autonomous vehicle lane-changing trajectory prediction based on a heuristic attention-aided encoder-decoder model. Transp. Res. C 140, 103706.
Williams, R.J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. 8 (3), 229–256.
Wu, T., Zhou, P., Liu, K., Yuan, Y., Wang, X., Huang, H., Wu, D.O., 2020. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. IEEE Trans. Veh. Technol. 69 (8), 8243–8256.
Xie, D.-F., Fang, Z.-Z., Jia, B., He, Z., 2019. A data-driven lane-changing model based on deep learning. Transp. Res. C 106, 41–60.
Xing, Y., Lv, C., Wang, H., Cao, D., Velenis, E., 2020. An ensemble deep learning approach for driver lane change intention inference. Transp. Res. C 115, 102615.

Yao, W., Zhao, H., Bonnifait, P., Zha, H., 2013. Lane change trajectory prediction by using recorded human driving data. In: 2013 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 430–436.
Ye, Y., Zhang, X., Sun, J., 2019. Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment. Transp. Res. C 107, 155–170.
Yu, C., Ni, A., Luo, J., Wang, J., Zhang, C., Chen, Q., Tu, Y., 2022. A novel dynamic lane-changing trajectory planning model for automated vehicles based on reinforcement learning. J. Adv. Transp. 2022.
Yu, S., Shi, Z., 2015. The effects of vehicular gap changes with memory on traffic flow in cooperative adaptive cruise control strategy. Phys. A 428, 206–223.
Yu, H., Tseng, H.E., Langari, R., 2018. A human-like game theory-based controller for automatic lane changing. Transp. Res. C 88, 140–158.
Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M.J., et al., 2016. Apache spark: a unified engine for big data processing. Commun. ACM 59 (11), 56–65.
Zhang, H., 2003. Driver memory, traffic viscosity and a viscous vehicular traffic flow model. Transp. Res. B 37 (1), 27–41.
Zhang, J., Chang, C., Zeng, X., Li, L., 2022. Multi-agent DRL-based lane change with right-of-way collaboration awareness. IEEE Trans. Intell. Transp. Syst. 24 (1), 854–869.
Zhang, X., Sun, J., Qi, X., Sun, J., 2019. Simultaneous modeling of car-following and lane-changing behaviors using deep learning. Transp. Res. C 104, 287–304.
Zhang, C., Zhu, J., Wang, W., Xi, J., 2021. Spatiotemporal learning of multivehicle interaction patterns in lane-change scenarios. IEEE Trans. Intell. Transp. Syst..
Zhao, D., Guo, Y., Jia, Y.J., 2017. Trafficnet: An open naturalistic driving scenario library. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1–8.
Zhao, J., Knoop, V.L., Wang, M., 2020. Two-dimensional vehicular movement modelling at intersections based on optimal control. Transp. Res. B 138, 1–22.
Zheng, Z., 2014. Recent developments and research needs in modeling lane changing. Transp. Res. B 60, 16–32.
Zheng, Z., Ahn, S., Chen, D., Laval, J., 2011. Applications of wavelet transform for analysis of freeway traffic: Bottlenecks, transient traffic, and traffic oscillations. Transp. Res. B 45 (2), 372–384.
Zheng, Z., Ahn, S., Chen, D., Laval, J., 2013. The effects of lane-changing on the immediate follower: Anticipation, relaxation, and change in driver characteristics. Transp. Res. C 26, 367–379.
Zheng, Z., Ahn, S., Monsere, C.M., 2010. Impact of traffic oscillations on freeway crash occurrences. Accid. Anal. Prev. 42 (2), 626–636.
Zhou, B., Wang, Y., Yu, G., Wu, X., 2017. A lane-change trajectory model from drivers' vision view. Transp. Res. C 85, 609–627.
Zhu, M., Du, S.S., Wang, X., Pu, Z., Wang, Y., et al., 2022. Transfollower: Long-sequence car-following trajectory prediction through transformer. arXiv preprint arXiv:2202.03183.
Zhu, M., Wang, Y., Pu, Z., Hu, J., Wang, X., Ke, R., 2020. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. Transp. Res. C 117, 102662.
Zhu, M., Wang, X., Wang, Y., 2018. Human-like autonomous car-following model with deep reinforcement learning. Transp. Res. C 97, 348–368.