# A Deep Reinforcement Learning Based Car Following Model for Electric Vehicle

**Yuankai Wu[1], Huachun Tan[2], Jiankun Peng[3], Bin Ran[4]**

1 School of Mechanical Engineering, Beijing Institute of Technology, No 5 yard, South Zhongguanchun Road, Beijing, China
2 School of Transportation Engineering, Southeast University, Sipailou 2, Nanjing, Jiangsu, China
3 School of Mechanical Engineering, Beijing Institute of Technology, No 5 yard, South Zhongguanchun Road, Beijing, China
4 Department of Civil and Environmental Engineering, University of Wisconsin-Madison,1415 Engineering Drive, Madison, WI, USA

## Abstract

Car following (CF) models are an appealing research area because they fundamentally describe longitudinal interactions of vehicles on the road, and contribute significantly to an understanding of traffic flow. There is an emerging trend to use data-driven method to build CF models. One challenge to the data-driven CF models is their capability to achieve optimal longitudinal driven behavior because a lot of bad driving behaviors will be learnt from human drivers by the supervised learning manner. In this study, by utilizing the deep reinforcement learning (DRL) techniques trust region policy optimization (TRPO), a DRL based CF model for electric vehicle (EV) is built. The proposed CF model can learn optimal driving behavior by itself in simulation. The experiments on following standard driving cycle show that the DRL model outperforms the traditional CF model in terms of electricity consumption.

**Keywords:** *autonomous electric vehicle, car following model, deep reinforcement learning, trust region policy optimization*

## 1. Introduction

Electric vehicle (EV) and autonomous vehicle (AV) are two flourishing technologies which would promote environment sustainability and improve community livability. Accomplish of the high level autonomous electric vehicle (AEV) requires breakthrough in numerous technology, among which longitudinal dynamics control is an unquestionable key factor gearing up the safety and efficiency of AEV.

In the transportation research field, car-following models have been successfully applied to describe longitudinal driving behavior under car following (CF) scenario. In order to analyze traffic flow in simulation program as simple as possible, researchers are often interested in describing CF behavior with mathematical models (Chandler et.al 1958, Maerivoet & De Moor, 2005). Although mathematical model based CF models are powerful and useful tools for analysis of driving behavior, there still require significant improvements. First, a calibration process is required for most models before they are able to analyze and simulate real traffic dynamics. Calibration, however, is a onerous process needed further studies (Punzo et.al, 2012). Second, the dynamics between driving environment and CF decision is very complex. A simple mathematical model is not able to fully model the correlation between environment and decision.

Recent developments in the field of big data have led to an interest in development of data-driven CF models (He et.al, 2015). Neural networks (NNs), a learning system with universal approximation ability, have been extensively used to describe CF behaviors because their ability to imitate human learning process from data. For example, Chong et.al, (2013) attempts to use fuzzy NNs to achieve data-driving CF models. Hongfei et.al, (2003) shows that the NNs models could accurately describe the following behavior of a driver after the training course on field data. Several studies (Khodayari et.al, 2012, Zheng et.al, 2013) have incorporated reaction delay into NNs models.

In recent years, deep learning (NNs with deep structure) have won numerous contests in pattern recognition and machine learning (Schmidhuber, 2015). To better model the CF behavior, researches have been using deep learning technique. The deep recurrent neural networks (RNNs) are the most popular deep learning structure for model CF because of its capability to model memory effect in processing sequential data. Wang et.al, (2018) have recently developed a deep RNN model for the representation of memory effect in the CF model, it is reported that their model achieves higher prediction accuracy than shallow NNs based models. Similar researches can be found in (Zhou et.al, 2017, Huang et.al, 2018).

The above studies especially the deep learning approaches demonstrate great flexibility of neural networks for modeling

CF behavior. However, these neural networks are trained in a supervised learning manner using real drivers' trajectories. Create a datasets with perfect CF trajectories might be expensive and unfeasible. The human driver itself is not perfect, whose driving behavior is limited by reaction delay, bad temper and so on. Therefore a CF model trained by supervised learning has inevitably learned a lot of bad driving behaviors from imperfect drivers, therefore is difficult to provide the best CF behavior. In problems such as Go (Silver et.al, 2016) and computer games (Mnih et.al, 2015), reinforcement learning (RL) successfully address these challenges. The essence of RL is learning through interaction. RL agent interacts with its environment and, upon observing the consequences of its actions, can learn to alter its own behaviour in response to rewards received (Arulkumaran et.al, 2017). A RL agent can theoretically achieve behavior that maximizes cumulative reward.

Deep learning has greatly enhance RL, with the exploitation of deep learning algorithms within RL defining the field of"deep reinforcement learning" (DRL). There are numerous DRL approaches including deep Q networks (DQN) (Mnih et.al, 2015), Evolutionary Strategy (ES) (Salimans et.al, 2017) and various policy gradient methods, such as TRPO (Schulman et.al, 2015), A3C (Mnih et.al, 2016), DDPG (Lillicrap et.al, 2015) and PPO (Schulman et.al, 2017). Those algorithms hold great promise for learning to solve challenging decision make problems such as CF.

The goal of this paper is to utilize DRL to achieve economic and safe longitudinal driving for AEV. First, we build a simulation model for electric vehicle using real-world data. Then we formulate the CF model as a markovian decision process (MDP) and propose to use DRL algorithm to solve the MDP. Finally we conduct simulated experiments on the new european driving cycle (NEDC), the results show that the DRL based CF approach is more energy-efficient than conventional CF models.

## 2. The Simulation Model For Electric vehicle

The EV powertrain and appearance of Roewe E50 are shown in Figure 1. The powertrain is composed of a drive motor and a power battery. The efficiency map of the traction motor is given in Figure 2. The main parameters of Roewe E50 are listed in Table 1.

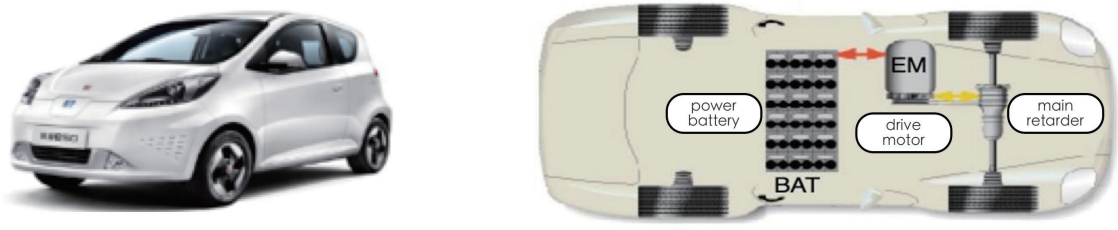


**Figure 1** The appearance and powertrain of Roewe E50

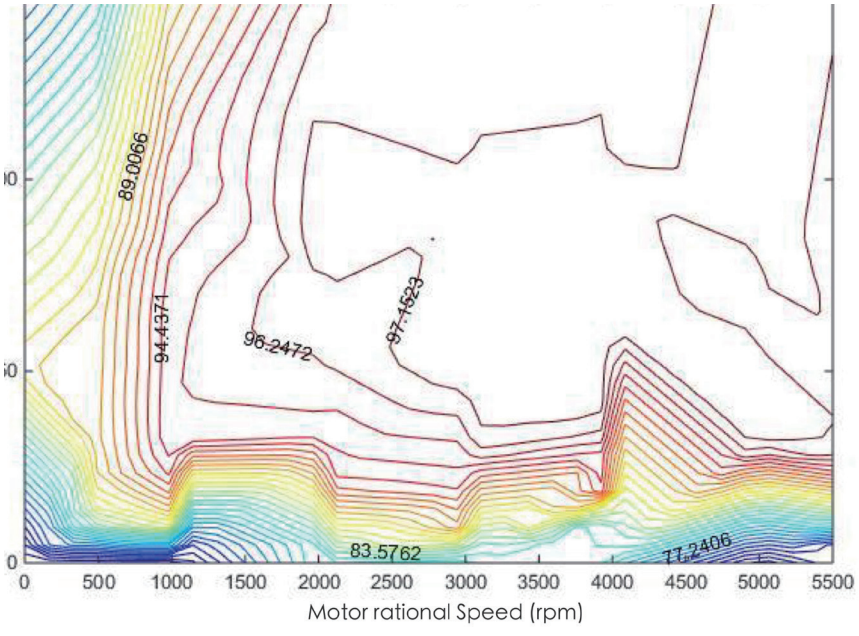


**Figure 2** The efficiency map of traction motor for Roewe E50

**Table 1** Main parameters of Roewe E50 specification

| Parameters | Value | Parameters | Value |
|---|---|---|---|
| Full mass | 1391.2kg | Air drag coefficient | 0.34 |
| Windward area | 1.83m$^2$ | Main reducer ratio | 6.2 |
| Rolling resistance coefficient | 0.0011 | Battery capacity | 77.7 Ah |
| Wheel radius | 0.262m | Peak velocity | 130 km/h |

There are two approaches to simulate the powertrain: the backward and forward approaches (Onori et.al, 2016). The backward approach is chosen in this paper. In a backward simulator, no driver model is necessary, the desired speed is a direct input to the simulator, and the energy consumption is output. The tractive force of the vehicle is calculated by:

$$F_{tr} = \delta ma + F_g + F_r + F_{AD}. \tag{1}$$

where $\delta$ is the rotational inertia coefficient, α is the acceleration of the vehicle, $F_g = mgsin\theta$,$F_r$ is the rolling resistance force $F_r = mgcos\theta C_r$, $C_r$ is the rolling resistance coefficient. $F_{AD} = 0.5p_aC_{AD}A_fv^2$, $p_a$ is the air density, $C_{AD}$ is the air drag coefficient, $A_f$ is the windward area, $v$ is the velocity of the vehicle.

The torque, rational speed and power of vehicle demand is computed by

$$T_{wh} = F_{tr}r_{wh},\ W_{wh} = \frac{v}{r_{wh}},\ P_{wh} = T_{wh}W_{wh}. \tag{2}$$

where $r_{wh}$ is the Wheel radius. The power of the motor is $P_{mot} = P_{wh}/\eta_d$. $\eta_d$ is the transmission efficiency. The power of the battery is then computed by $P = P_{mot}/\eta_{mot}$. $\eta_{mot}$. $\eta_{mot}$ is the efficiency of the motor, which is computed by the efficiency map given in Figure 2.

Then the battery current can be computed by

$$I = \frac{V_{oc} - \sqrt{V_{oc}^2 - 4RP}}{2R}. \tag{3}$$

$V_{oc}$ is the open-circuit voltage of the battery, and $R$ is the internal resistance of battery. Then the consume of SOC of the battery can be computed by

$$\Delta SOC = \frac{It}{Q}. \tag{4}$$

where Q is the battery capacity. To make sure the simulation model for EV is accurate, we compare the battery current of simulation model and real vehicle using a real trajectory. The result is given in Figure 3. The results show that the battery current curve of simulation model is close to the one of real vehicle.

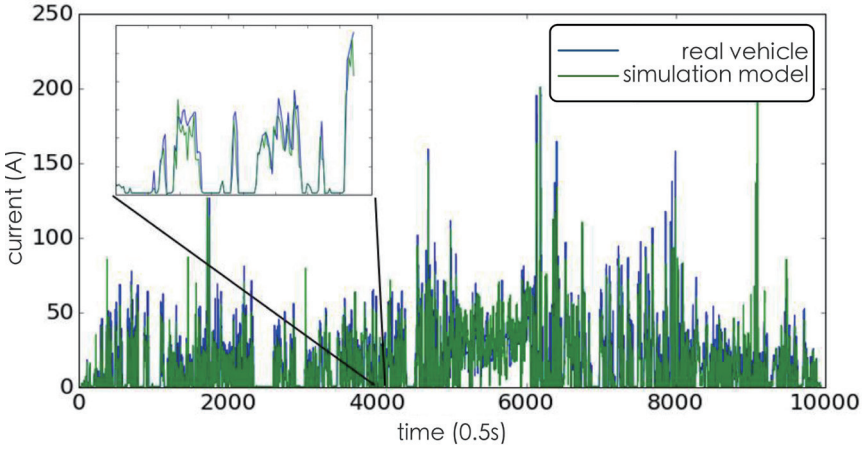


**Figure 3** The comparison between battery current of real vehicle and simulation model

## 3. Car Following As a markovian Decision Process

In this paper, we consider the problem of CF trajectory planning for AEV. The goal of the planning is to minimize the electricity consumption and simultaneously guarantee the safety and effectiveness of CF. Driving speed and gap of the leader are collected real-time by V2V devices and/or sensors of following vehicle. An illustration of the problem is available in Figure 4. The aim of CF trajectory planning is to decide the real-time acceleration of the follower.

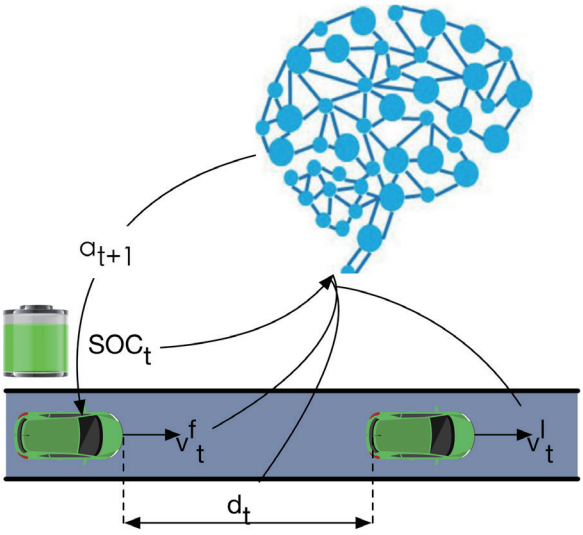


**Figure 4** The CF problem in this paper

To tackle this problem, we propose to formulate the problem using markovian decision process (MDP), which is defined as $\mathfrak{G} = (S, A, P, R, \gamma)$, where $S, A, P, R, \gamma$ are the sets of states, action space, transition probability functions, reward functions, and a discount factor respectively. The definitions are given as follows:

State $s_t \in S$: The state of the CF problem is defined as a vector:

$$s_t = \left(SOC_t, d_t, {v^l}_t, {v^f}_t\right). \tag{5}$$

where $SOC_t$ is the state of charge of follower's battery at time $t, d_t, {v^l}_t, {v^f}_t$ are the distance, speed of the leader, and speed of the follower at time $t$. The reason we choose $SOC_t$ as a state variable is that the electricity consumption is impacted by the $SOC$ of the battery.

Action $a_t \in A$: The action of the following vehicle is defined as its action. The acceleration is ranging from -3 to 3.

Reward function $r_t \in R = S \times A \to \mathcal{R}$: Since the follower should follow with the leader with an appropriate distance while save the electricity consumption, both the distance and electricity consumption should be taken into account in modelling reward. The reward is defined as the sum cost of the distance reward and electricity consumption as follows:

$$r_t = f_r\left(d_t, {v^f}_t\right) - Q\Delta SOC_t Price_e \tag{6}$$

where $f_r()$ is a function that mapping $d_t, {v^f}_t$ into the distance reward. $Price_e$ is the RMB cost for electricity. $f_r()$ is defined as:

$$f_r\left(d_t, {v^f}_t\right) = \begin{cases} -0.0035\left(0.1{v^f}_t - d_t\right), \ if \ 0.1 < d_t < 0.1{v^f}_t \\ -0.00175\left(d_t - {v^f}_t\right) \quad if \ d_t > {v^f}_t \\ -1 \quad if \ d_t \le 0.1 \end{cases}. \tag{7}$$

The distance reward function given in Eq (7) means that when the distance between leader and follower is below the safety distance the function outputs a low reward value. Moreover, the function also outputs a low reward value if the follower are too far away from the leader.

State transition probability $p(s_{t+1}|s_t, a_t)$: $S \times A \times S \to [0:1]$: It gives the probability of transiting to $s_{t+1}$ given a action $a_t$ is taken in the current state $s_t$. The transition of $SOC_t$ in Eq (5) is determined by the EV simulation model given in section 2. ${v^f}_t$ of state in Eq (5) is computed by ${v^f}_{t+1} = {v^f}_t + a_t$. $d_t$ and ${v^l}_t$ are determined by the future driving behaviour of the leader.

## 4. Trust Region Policy Optimization

The essence of DRL is to use deep learning techniques to search an optimal action policy for MDP. In this paper, we use a specific policy gradient method — Trust Region Policy Optimization (TRPO) to obtain CF policies. We introduce the basic principle of TRPO in this section.

In RL, we optimize a policy $\pi_\theta$ for the maximum expected discounted rewards:

$$\max_{\theta} J(\pi_\theta) = E_{r\sim\pi_\theta}(\sum_{t=0}^{\infty}\gamma^t r_t). \tag{8}$$

The policy gradient (PG) computes the steepest ascent direction for the rewards and update the policy towards that direction.

$$g = \nabla_\theta J(\pi_\theta), \quad \theta_{k+1} = \theta_k + \alpha g. \tag{9}$$

A problem with PG is that improper learning rate $\alpha$ will cause vanishing or exploding gradient. Moreover, $J(\pi_\theta)$ is sensitive to noise or function approximation error. In order to make the optimization more robust, an advantage function is defined:

$$\begin{aligned} &Q_\pi(s_t, a_t) = E_{s_{t+1},a_{t+1},\dots}\left(\sum_{t=0}^{\infty}\gamma^l r_{t+l}\right), \ V_\pi(s_t) = E_{a_t,s_{t+1},\dots}\left(\sum_{t=0}^{\infty}\gamma^l r_{t+l}\right), \\ &A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s). \end{aligned} \tag{10}$$

The advantage function $A_\pi()$ describes how good the action $a$ is compared to the average of all the action. Using the advantage function, the loss function of policy gradient becomes

$$\max_{\theta} J(\pi_\theta) = E_t(\log \pi_\theta(a_t|s_t)A_t). \tag{11}$$

In order to solve the problem using Monte Carlo simulation, the importance sampling method is applied to transform the loss function into the following form:

$$\max_{\theta^{old}} J(\pi_\theta) = E_t\left(\log \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta^{old}}(a_t|s_t)}A_t\right) \tag{12}$$

It is suggested that by maximize the following loss function, we are guaranteed to improve the policy:

$$\max_{\theta^{old}} \ E_t\left(\log \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta^{old}}(a_t|s_t)}A_t\right) - \beta E_t[KL[\pi_\theta(\cdot\,|s_t), \pi_{\theta^{old}}(\cdot\,|s_t)]] \tag{13}$$

where $KL(\cdot,\cdot)$ is KL divergence. The detail about the mathematical derivation of TRPO can be found in (Schulman et.al, 2015).

TRPO can be implemented via Actor-Crtic architecture, the advantage value $\mathrm{A_t}$ can be estimated by the critic when training actor's parameters $\theta$. Both actor and critic can be parameterized by neural networks using parameters $\epsilon$ and $\theta$. The critic parameters $\epsilon$ is learnt using the gradients from the TD error signal:

$$\min_{\epsilon}(r_t + \gamma V^\epsilon(s_{t+1}) - V^\epsilon(s_t))^2 \tag{14}$$

The advantage function $A_t$ is estimated by $V^\epsilon(s_t) - (r_t + \gamma V^\epsilon(s_{t+1}))$. The algorithm for TRPO based CF is summarized in Algorithm 1.

Algorithm 1 Framework of TRPO for CF

1.Randomly initialize critic and actor network with parameters $\epsilon$ and $\theta$;
2.For episode = 1 to m do
3. Receive initial observation state $s_0 = (SOC_0, d_0, {v^l}_0, {v^f}_0)$;
4. For t = 1 to time length of following T do
5. Select action $a_t$ according to the current actor;
6. Execute action $a_t$ and observe reward $r_t$, new state $s_{t+1}$ using simulation EV model;
7. Update actor parameters $\theta$ by maximizing the loss in (13);
8. Update critic parameters $\epsilon$ by minimizing the loss in (14);
9. end for
10. end for

## 5. Experimental Results

In this section, we present the quantitative and qualitative experiment results on following a vehicle driving with standard cycle NEDC. Their initial distance is 2m. The actor and the critic of the agent are composed of neural networks. Specifically, the actor $f^\theta()$ is expressed by:

$$h^a = relu(W_h{}^a s + b_h{}^a), a^\mu = 3\tanh\left(W_\mu h^a + b_\mu\right), a^\sigma = \mathrm{softplus}(W_\sigma h^a + b_\sigma) \tag{15}$$

where relu, tanh and softplus are nonlinear activations. $W_h{}^a \in R^{200\times 4}$ and $b_h{}^a \in R^{200\times 1}$ are the parameters for hidden layers of actor. $W_\mu \in R^{1\times 200}$ and $b_\mu \in R^{1\times 1}$ are the parameters for mean value of action. $W_\sigma \in R^{1\times 200}$ and $b_\sigma \in R^{1\times 1}$ are the parameters for variance of action. The acceleration of the vehicle is sampled from Gaussian distribution $N(\mu, \sigma^2)$. The

critic $f^{\epsilon}()$ is expressed by:

$$h^c = relu(W_h{}^c s + b_h{}^c), v = linear(W_v h^c + b_v) \quad (16)$$

where relu and linear are nonlinear activations. $W_h{}^c \in R^{200\times1}$ and $b_h{}^c \in R^{200\times1}$ are the parameters for hidden layers of critic. $W_v \in R^{1\times200}$ and $b_v \in R^{1\times1}$ are the parameters for value function estimation of critic. The actor and critic are built upon Tensorflow (https://www.tensorflow.org/).

The learning process of TRPO on CF is given in Figure 5. The agent achieves progressively high reward. It is observed that the agent is learnt to drive far behind the leader at the beginning because the electricity cost is very low if the vehicle drives very slow. Then the distance reward $f_r(d_t, v^f{}_t)$ motivates the agent to drive behind the leader closely, the average distance goes down and the RMB cost of electricity plateau at 1.3 ￥.

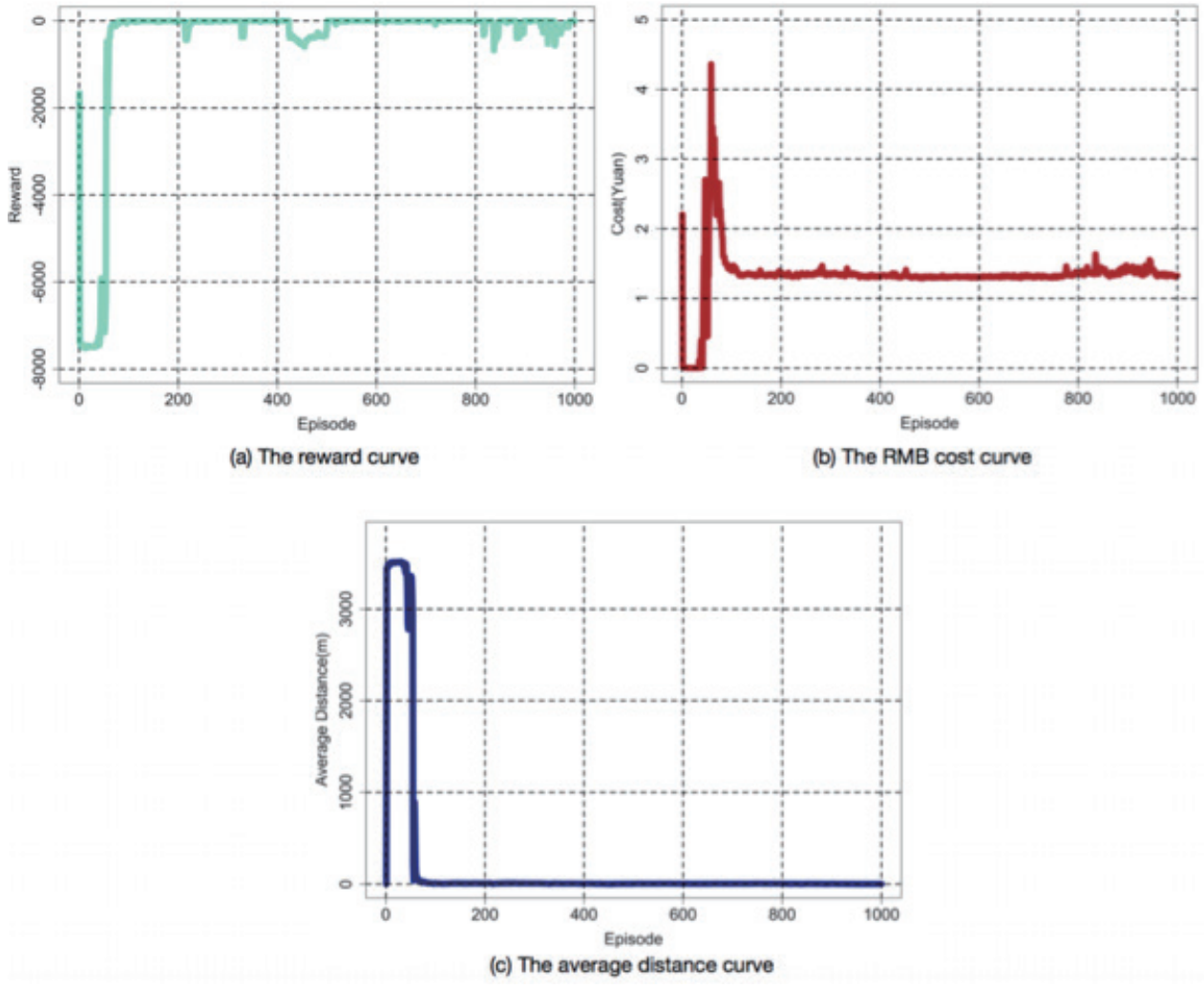


**Figure 5** The learning process of TRPO on CF

We compare the TRPO based CF model with several conventional CF models including Krauß, SmartSK, and Wiedemann models. Those models are well built-in SUMO- an open source traffic simulation software. The software is highly flexible, well documented and supports set the speed limits for each lane using its API--the Traffic Control Interface (TraCI) package. In SUMO, we set the leader driving with the speed from NEDC driving cycle, and the followers with different car following models are driving behind the leader. The initial distance between leader and follower is set to 2m. The minimum gap between leader and follower is set to 0.1m. The initial SOC of the simulated Roewe E50 is set to 0.85. The final SOC and RMB cost of the electricity consumption of different models are given in Table 2. It is obvious that the DRL based TRPO model outperforms other conventional CF models. The reason is that the DRL based CF model utilizes neural networks to model the CF behaviour, which has more powerful capability than model with simple mathematical equation. Moreover, the DRL based CF is trained by a reward-driven manner, thus can achieve optimal electricity

consumption and following distance using the reward signal given in Eq (6).

**Table 2** The final SOC and RMB cost of different CF models

| Method | Final_SOC | RMB cost |
|---|---|---|
| TRPO | 0.7972 | 1.3261 |
| Krauss | 0.7833 | 1.6746 |
| SmartSK | 0.7210 | 3.2341 |
| Wiedemann | 0.7892 | 1.5296 |

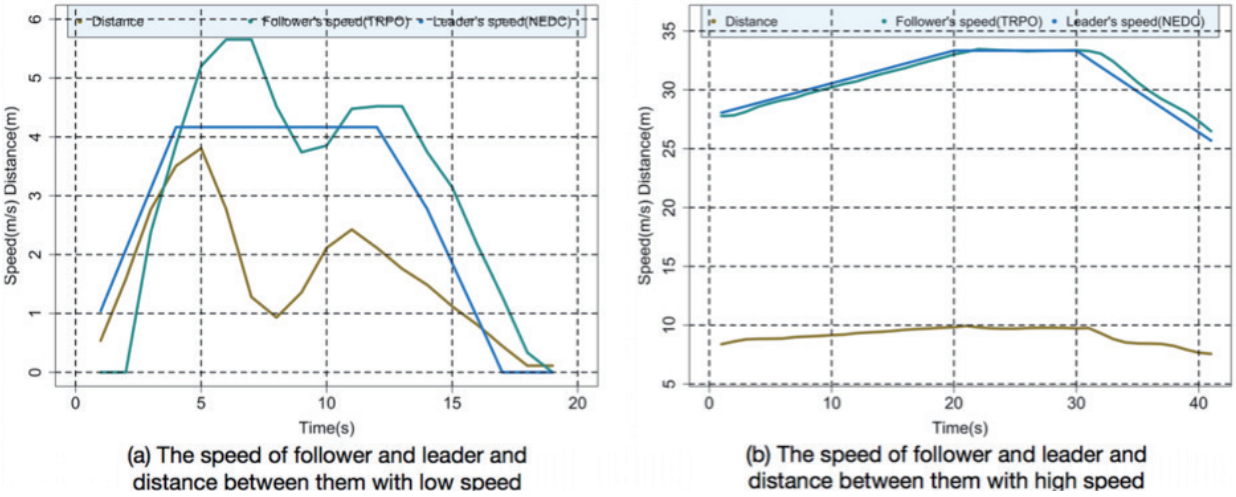


**Figure 6** The speed curves of follower (TRPO), leader (NEDC) and corresponding distance between them under two different cases

Figure 6 plots the speed curves of TRPO based follower and its distance behind the leader driving with NEDC driving cycle under two cases. From the plots, we can find that the TRPO based follower can follow the leader with a very low distance. In figure (b), the distance between the leader and follower is under 10m even the two vehicles are driving with speed above 30m/s. We can observe a 2s acceleration//deceleration duration of follower when its leader starts acceleration/deceleration. The duration does not deteriorate the driving safety, the follower can precisely control its speed and maintain the distance behind leader above 0.1m. The results indicate that the DRL based CF model is very effective. It ensures that the high speed AVs can drive with a very small gap. We can apply the DRL based CF model to autonomous platoon control, which could improve the traffic capacity. Moreover, it has been proved that the platoon with high speed and low distance can significantly reduce the air drag force of each vehicle, therefore leads to reduction of fuel consumption (Liang et.al, 2015). The proposed DRL based CF model can be further applied to longitudinal control of the platoon.

## 6. Conclusion And Future works

In this paper, we have investigated how to use DRL to CF problem of AEV with respect to electricity consumption and following distance. We build a energy consumption model of Roewe E50 using real-world data. The CF problem is formulated as a MDP, then it is solved by TRPO- a popular DRL framework. Lastly, we compared our DRL based CF approach with conventional CF models in following standard driving cycle NEDC, which suggest that our model is better than traditional models in terms of electricity consumption and following distance.

The obtained results have assumed only two vehicles interact with each other. However, in reality, traffic is commonly a significant factor in longitudinal control since it will affect the possibilities to form CF pairs and the potential electricity savings. Therefore, it is of interest to extend the DRL based driving behaviour control to more complex traffic environment. Furthermore, it is believed that the air drag force will be significantly reduced in a platoon. The application of DRL on platoon control will be a valuable future direction. Moreover, the aerodynamic forces modelling for simulation of DRL based driving control is also very important, which could tell us the optimal vehicle distance to reduce fuel/electricity consumption.

## 7. Aacknowledgements

The work was supported by national natural science foundation of China (61620106002 and 5170520). The authors

acknowledge the help of Renzong Lian, who has helped us to perform traffic simulation using SUMO. The parameters of Roewe E50 is provided by SAIC Motor.

## References

[1] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). A brief survey of deep reinforcement learning. arXiv preprint arXiv:1708.05866.
[2] Chandler, R. E., Herman, R., & Montroll, E. W. (1958). Traffic dynamics: studies in car following. Operations research, 6(2), 165-184.
[3] Chong, L., Abbas, M. M., Flintsch, A. M., & Higgs, B. (2013). A rule-based neural network approach to model driver naturalistic behavior in traffic. Transportation Research Part C: Emerging Technologies, 32, 207-223.
[4] He, Z., Zheng, L., & Guan, W. (2015). A simple nonparametric car-following model driven by field data. Transportation Research Part B: Methodological, 80, 185-201.
[5] Hongfei, J., Zhicai, J., & Anning, N. (2003, October). Develop a car-following model using data collected by" five-wheel system". In Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE (Vol. 1, pp. 346-351). IEEE.
[6] Huang, X., Sun, J., & Sun, J. (2018). A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. Transportation Research Part C: Emerging Technologies, 95, 346-362.
[7] Khodayari, A., Ghaffari, A., Kazemi, R., & Braunstingl, R. (2012). A modified car-following model based on a neural network model of the human driver effects. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 42(6), 1440-1449.
[8] Liang, K. Y., Mårtensson, J., & Johansson, K. H. (2016). Heavy-duty vehicle platoon formation for fuel efficiency. IEEE Transactions on Intelligent Transportation Systems, 17(4), 1051-1061.
[9] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.
[10] Maerivoet, S., & De Moor, B. (2005). Cellular automata models of road traffic. Physics reports, 419(1), 1-64.
[11] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529.
[12] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016, June). Asynchronous methods for deep reinforcement learning. In International conference on machine learning (pp. 1928-1937).
[13] Onori, S., Serrao, L., & Rizzoni, G. (2016). Hybrid electric vehicles: Energy management strategies. Berlin Heidelberg: Springer.
[14] Punzo, V., Ciuffo, B., & Montanino, M. (2012). Can results of car-following model calibration based on trajectory data be trusted?. Transportation Research Record, 2315(1), 11-24.
[15] Salimans, T., Ho, J., Chen, X., Sidor, S., & Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. arXiv preprint arXiv:1703.03864.
[16] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.
[17] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In International Conference on Machine Learning (pp. 1889-1897).
[18] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
[19] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), 484.
[20] Wang, X., Jiang, R., Li, L., Lin, Y., Zheng, X., & Wang, F. Y. (2018). Capturing car-following behaviors by deep learning. IEEE Transactions on Intelligent Transportation Systems, 19(3), 910-920.
[21] Zheng, J., Suzuki, K., & Fujita, M. (2013). Car-following behavior with instantaneous driver–vehicle reaction delay: A neural-network-based methodology. Transportation research part C: emerging technologies, 36, 339-351.
[22] Zhou, M., Qu, X., & Li, X. (2017). A recurrent neural network based microscopic car following model to predict traffic oscillation. Transportation research part C: emerging technologies, 84, 245-264.