Letter

Towards Energy-Efficient Autonomous Driving: A Multi-Objective Reinforcement Learning Approach

Xiangkun He and Chen Lv, *Senior Member, IEEE*

Dear Editor,

With the development of automobile industry and artificial intelligence (AI) domains, autonomous vehicles (AVs) are becoming a reality and promise to revolutionize human mobility [1]–[3]. The decision-making system of AVs is crucial, which is typically required to trade off multiple competing objectives. For example, when determining driving policies, autonomous electric vehicles (AEVs) need to consider two conflicting objectives: transport efficiency and electricity consumption. As one of state-of-the-art AI technologies, reinforcement learning (RL) has demonstrated its potential in a series of challenging tasks. Accordingly, RL has attracted considerable attention from global researchers [4].

Many studies have employed various RL methods to learn the optimal driving modes (e.g., keeping, acceleration and deceleration) of AVs. For instance, an entropy-constrained RL approach is developed to enable AEVs to learn multi-modal driving policies in [5].

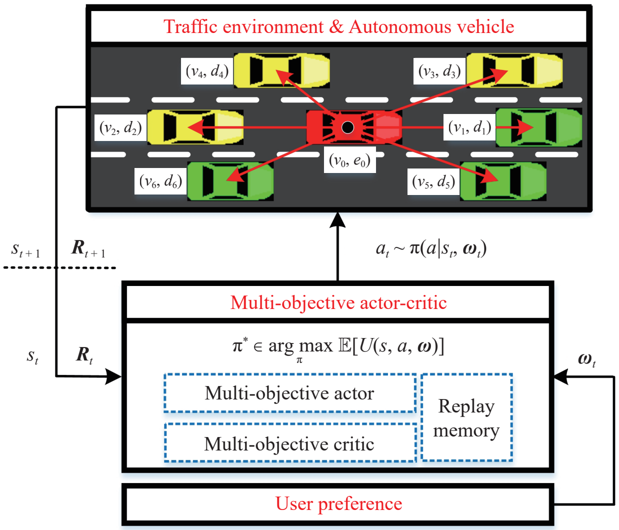
While the existing methods have achieved numerous compelling results, there are still various technical barriers to AVs. Firstly, many real-world decision-making tasks have to trade off multiple conflicting objectives. Secondly, user preferences regarding multiple objectives may change with driving conditions. For example, passengers of AEVs typically focus more on travel efficiency than on energy conservation. In contrast, when AEVs are about to run out of energy, users put more weight on energy saving. Hence, we should attach importance to the multi-objective decision-making problem in AVs. Several existing studies have attempted to tackle this challenge. One popular scheme is to leverage a hierarchical architecture considering personalized driving behaviours and multi-objective cost function [6] and [7]. In such a method, the decision module is designed by combining game theory or potential field with personalized driving behaviours. Then, the control module is developed via model predictive control (MPC) based on the multi-objective cost function.

The aforementioned approach is highly interpretable; however, some challenges remain. First, such an approach is computationally tricky for large state spaces, and its generalization to unseen cases is intractable to guarantee, since it requires solving an optimization problem in each of the different conditions. Second, such an approach cannot directly adapt to arbitrary user preferences, as it has to solve a new optimization problem based on a different multi-objective cost function when user preferences change. Consequently, to address these requirements, the developed solution should satisfy two aspects. On the one hand, learning-based methods are necessary to allow trained policy models to handle previously unseen scenarios without learning. On the other hand, the input to the policy model is required to include user preferences concerning multiple objectives, which aims to enable the trained policy model to directly approximate the optimal driving policy based on the current state and user preferences.

Unlike traditional RL that optimizes policies via a single scalar reward, multi-objective RL (MORL) seeks to learn the Pareto optimal policies through combining a multi-objective reward vector with user preferences. Although a small number of researchers have employed MORL to cope with autonomous driving tasks [8] and [9], they cannot provide user-preference-conditioned driving policies over the entire preference space by a single model. Furthermore, MORL techniques have not been fully explored in energy-aware autonomous driving.

Here, a novel MORL approach, called multi-objective actor-critic (MOAC), is proposed for user-preference-conditioned decision-making that trades off the energy consumption and travel efficiency of AEVs. The MOAC algorithm tries to train a single model that approximates Pareto optimal policies over the entire user preference space. Specifically, the proposed MOAC method maximizes the dot product between a user preference and a vectorized action-value function, enabling the agent to learn Pareto optimal policies. At the same time, to ensure the diversity of learned policies and the efficient alignment between user preferences and corresponding optimal policies, the MOAC algorithm minimizes the norm of the cross-product between the user preference and the vectorized action-value function. The three stochastic dynamic traffic flows with different densities are carried out to assess the performance of the proposed method in highway scenarios via simulation of urban mobility (SUMO) [10]. The results demonstrate that our method is effective and surpasses both classical and state-of-the-art baselines.

Proposed methodology: Fig. 1 illustrates our user-preference-conditioned decision-making framework of AEVs in highway scenarios. s_t, $\boldsymbol{\omega}_t$, a_t, $\boldsymbol{R}_t$ and $\pi(a|s_t,\boldsymbol{\omega}_t)$ represent the state, user preference, action, multi-objective reward vector and user-preference-conditioned policy at the time step t, respectively. Moreover, π^* and $U(\cdot)$ indicate the optimal policy and utility function, respectively. Ego vehicle is red, and it is an AEV. The vehicles of other colors are social vehicles. The input of the agent contains 14 dimensions, and the detailed description is shown in Fig. 1. Its output is the continuous longitudinal acceleration or deceleration of the ego vehicle.



v_0 (m/s): Velocity of the ego vehicle. e_0 (Wh/s): Energy consumption of the ego vehicle.
v_1 (m/s): Velocity of the vehicle in front in same lane. d_1 (m): Distance from the vehicle in front in same lane.
v_2 (m/s): Velocity of the vehicle behind in same lane. d_2 (m): Distance from the vehicle behind in same lane.
v_3 (m/s): Velocity of the vehicle in front in left lane. d_3 (m): Distance from the vehicle in front in left lane.
v_4 (m/s): Velocity of the vehicle behind in left lane. d_4 (m): Distance from the vehicle behind in left lane.
v_5 (m/s): Velocity of the vehicle in front in right lane. d_5 (m): Distance from the vehicle in front in right lane.
v_6 (m/s): Velocity of the vehicle behind in right lane. d_6 (m): Distance of the vehicle behind in right lane.

Fig. 1. Illustration of the proposed user-preference-conditioned decision-making approach for AEVs in a highway scenario.

Corresponding author: Chen Lv.

Citation: X. K. He and C. Lv, "Towards energy-efficient autonomous driving: A multi-objective reinforcement learning approach," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 5, pp. 1329–1331, May 2023.

The authors are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: xiangkun.he@ntu.edu.sg; lyuchen@ntu.edu.sg)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123378

To learn the optimal user-preference-conditioned policies, we design the utility function $U(\cdot)$ as

$$U(s,a,\boldsymbol{\omega}) = \boldsymbol{\omega}^T \cdot \boldsymbol{Q}^{\pi}(s,a,\boldsymbol{\omega}) - \alpha \left\| \boldsymbol{\omega}^T \times \boldsymbol{Q}^{\pi}(s,a,\boldsymbol{\omega}) \right\| \tag{1}$$

where α is a weight. $\boldsymbol{\omega} = [\omega_1,\ldots,\omega_k]^T$ represents user preference vector. ω_k corresponds to a preference across one objective. k denotes the objective number. The multi-objective reward function vector can be represented as $\boldsymbol{R}(s,a) = [r_1,\ldots,r_k]^T$. The reward r_k is scalar. Moreover, a multi-objective action-value function can be expressed as $\boldsymbol{Q}^{\pi}(s,a,\boldsymbol{\omega}) = [Q_1^{\pi},\ldots,Q_k^{\pi}]^T$, where the action-value function for objective k is represented as $Q_k^{\pi}(\cdot) = \mathbb{E}[\sum_t \gamma^t r_k(s_t,a_t)]$. Each $Q_k^{\pi}(\cdot)$ denotes one objective. Clearly, we expect $\boldsymbol{Q}^{\pi}(\cdot)$ to be parallel to the user preference $\boldsymbol{\omega}$ and its norm to be large.

Therefore, the Pareto optimal policy can be learned by solving the problem $\pi^* \in \arg\max_{\pi} \mathbb{E}[U(s,a,\boldsymbol{\omega})]$. The proposed MOAC comprises policy evaluation and improvement, and these two learning processes are optimized alternately until the policy converges.

Multi-objective policy evaluation: Since the target and current networks are too similar to provide an independent estimation in actor-critic, our MOAC algorithm employs two multi-objective action-value functions with network parameters ϕ^z, $z \in \{1,2\}$ to improve the performance of the policy model. Additionally, the loss function of multi-objective critic network can be defined as

$$J_c^1(\phi^z) = \mathop{\mathbb{E}}_{Ts\sim\mathcal{M}}\left[\left\| \boldsymbol{y} - \boldsymbol{Q}^{\pi}(s_t,a_t,\boldsymbol{\omega}_t;\phi^z) \right\|_2^2 \right] \tag{2}$$

where $\boldsymbol{y}$ represents the target vector, and Ts denotes transition sampled from replay memory $\mathcal{M}$.

The action with the parameterized deterministic policy and the random noises at time step t is represented as $\hat{\pi}(s_t,\boldsymbol{\omega}_t;\theta) = \pi(s_t,\boldsymbol{\omega}_t;\theta) + \beta\delta$, where θ is the parameter of multi-objective actor network, β is a weight, δ denotes random noises, $\delta \sim \mathcal{N}(0,1)$, and $\mathcal{N}(\cdot)$ denotes Gaussian distribution.

The minimum estimation between two target multi-objective action-value functions is utilized to train the multi-objective critic network. The target vector at time step t can be written as

$$\boldsymbol{y} = \boldsymbol{R}(s_t,a_t) + \gamma \min_{z\in\{1,2\}} \hat{\boldsymbol{Q}}^{\pi}(s_t,\hat{\pi}(s_t,\boldsymbol{\omega}_t;\theta),\boldsymbol{\omega}_t;\bar{\phi}^z) \tag{3}$$

where $\gamma \in (0,1)$ denotes a discount factor, $\hat{\boldsymbol{Q}}^{\pi}(s_t,a_t,\boldsymbol{\omega}_t;\bar{\phi}^z)$ indicates the target multi-objective action-value function with the network parameters $\bar{\phi}^z$.

To further consider user preference in the learning process, a scalar target $\bar{y}$ is developed as

$$\bar{y} = \boldsymbol{\omega}^T \boldsymbol{R}(s_t,a_t) + \gamma \min_{z\in\{1,2\}} \boldsymbol{\omega}^T \hat{\boldsymbol{Q}}^{\pi}(s_t,\hat{\pi}(s_t,\boldsymbol{\omega}_t;\theta),\boldsymbol{\omega}_t;\bar{\phi}^z). \tag{4}$$

The second loss function for multi-objective critic network can be designed as

$$J_c^2(\phi^z) = \mathop{\mathbb{E}}_{Ts\sim\mathcal{M}}\left[\left\| \bar{y} - \boldsymbol{\omega}^T \boldsymbol{Q}^{\pi}(s_t,a_t,\boldsymbol{\omega}_t;\phi^z) \right\|_2^2 \right]. \tag{5}$$

Hence, with (2) and (5), the parameters of multi-objective action-value functions can be updated by minimizing the loss function $J_c(\phi^z) = \frac{1}{2}\left(J_c^1(\phi^z) + J_c^2(\phi^z)\right)$. Furthermore, the parameters $\bar{\phi}^z$ of the target multi-objective action-value functions can be updated through Polyak averaging $\bar{\phi}^z \leftarrow \rho\bar{\phi}^z + (1-\rho)\phi^z$, where ρ is a scale factor between 0 and 1.

Multi-objective policy improvement: The policy improvement aims to optimize and update the policies. The average multi-objective action-value function $\bar{\boldsymbol{Q}}^{\pi}(s,a,\boldsymbol{\omega})$ is leveraged

$$\bar{\boldsymbol{Q}}^{\pi}(\cdot) = \frac{1}{2}[\boldsymbol{Q}^{\pi}(s,\hat{\pi}(s,\boldsymbol{\omega};\theta),\boldsymbol{\omega};\phi^1) + \boldsymbol{Q}^{\pi}(s,\hat{\pi}(s,\boldsymbol{\omega};\theta),\boldsymbol{\omega};\phi^2)]. \tag{6}$$

Hence, with (6), we can redefine the utility $U(\cdot)$

$$U(s,a,\boldsymbol{\omega}) = \boldsymbol{\omega}^T \cdot \bar{\boldsymbol{Q}}^{\pi}(s,a,\boldsymbol{\omega}) - \alpha \left\| \boldsymbol{\omega}^T \times \bar{\boldsymbol{Q}}^{\pi}(s,a,\boldsymbol{\omega}) \right\|. \tag{7}$$

The Pareto optimal policy can be solved by maximizing the following loss function of multi-objective actor network:

$$J_a(\theta) = \mathop{\mathbb{E}}_{Ts\sim\mathcal{M}}[\boldsymbol{\omega}^T \cdot \bar{\boldsymbol{Q}}^{\pi}(s,a,\boldsymbol{\omega}) - \alpha \left\| \boldsymbol{\omega}^T \times \bar{\boldsymbol{Q}}^{\pi}(s,a,\boldsymbol{\omega}) \right\|]. \tag{8}$$

Algorithm 1 Reward Function Vector for Two Objectives

```
Input: State and action of the MORL agent.
$r_1(s,a) = v_0/50$.                  \\* Encourage agent to be more efficiency
$r_2(s,a) = \max(0, 1 - e_0/100)$.    \\* Encourage energy conservation
if Collision occurs then
    $r_1(s,a) = r_1(s,a) - 0.50$.     \\* Penalize collision
    $r_2(s,a) = r_2(s,a) - 0.50$.     \\* Penalize collision
end if
Output: $\boldsymbol{R}(s,a) = (r_1(s,a), r_2(s,a))$.
```

Algorithm 1 outlines the reward function vector that involves two competing objectives. The first reward $r_1(s,a)$ corresponds to the objective concerning travel efficiency. The second reward $r_2(s,a)$ represents the objective regarding energy conservation. Furthermore, if the vehicle collides, it receives a penalty signal that penalizes the unsafe action. The MOAC scheme simultaneously optimizes these two conflicting objectives to enable the autonomous agent to learn user-preference-conditioned Pareto optimal policies.

Results and discussions: Our evaluation is implemented to test the performance of the proposed user-preference-conditioned decision-making approach for AEVs in highway scenarios. SUMO is adopted to create three highway scenarios with the stochastic mixed traffic flows based on different densities. *Pn*, *Pl* and *Ph* are leveraged to represent the probabilities for emitting a vehicle each second in stochastic traffic flows based on normal, low and high densities, respectively. We set *Pn*, *Pl* and *Ph* as 0.10, 0.20 and 0.30, respectively.

The MOAC algorithm and the baselines are assessed in both training and testing. The policy models of all the methods are trained and tested in the stochastic traffic flow with the normal density. Additionally, the mixed traffic flows with the low and high densities are only utilized to test the performance of the trained policy models.

To benchmark the proposed approach, we set up comparisons with the classical MORL algorithm with the weighted-sum scalarization [11] and the state-of-the-art envelope-MORL method [12]. Two classical MORL algorithms are implemented by combining the weighted-sum scalarization with the twin delayed deep deterministic (TD3) framework [13]. The first classical MORL method is called the scalar actor-based multi-objective TD3 (SA-MOTD3). The second classical MORL scheme is named scalar actor-critic-based multi-objective TD3 (SAC-MOTD3). The envelope multi-objective TD3 (EMOTD3) algorithm is developed as the state-of-the-art MORL baseline via combinin the envelope-MORL generalized framework with TD3.

Hypervolume is widely leveraged to measure the optimality and convergence of Pareto optimal policies or solutions. For model training phase, these four algorithms are assessed by five random seeds in the stochastic mixed traffic flows with the normal density. Fig. 2(a) demonstrates that the MOAC method outperforms the baselines with a large margin, both in terms of the hypervolume and the learning efficiency. The solid curve represents the mean, and the shaded region indicates the standard deviation. Moreover, we can find that, in contrast to the baselines, our MOAC algorithm converges rapidly and steadily to the optimal hypervolume during training.

The performance of the final policy models traind via SA-MOTD3, SAC-MOTD3, EMOTD3 and MOAC algorithms are evaluated in Table 1. To measure the hypervolume of the each policy, one policy model is tested via randomly sampling 500 preferences. The results demonstrate that our MOAC agent outperforms all the baselines in three highway scenarios based on the stochastic mixed traffic flows with different densities. For example, in comparison with SA-MOTD3, SAC-MOTD3 and EMOTD3 agents, the hypervolume based on the MOAC agent is enhanced by about 32.15%, 71.62% and 5.86% respectively, in the mixed traffic flows with high density.

Figs. 2(b) and 2(c) visually illustrates the quality of the Pareto fronts found by the single models trained via the different algorithms in the mixed traffic flows with low and high densities. It is obvious that the proposed MOAC technique can find more Pareto optimal

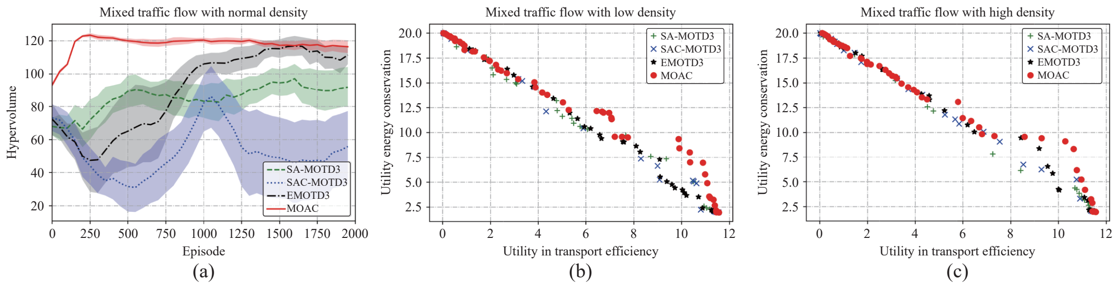


Fig. 2. Evaluation of our method and the baselines. (a) The learning curves of the different agents; (b) and (c) Pareto fronts obtained by the different models.

Table 1. Hypervolume of Different Agents in Different Traffic Flows

Method	Low density	Normal density	High density
SA-MOTD3	91.06 ± 37.17	104.83 ± 18.76	93.84 ± 28.06
SAC-MOTD3	85.29 ± 53.00	80.27 ± 55.16	72.26 ± 52.60
EMOTD3	118.00 ± 10.20	116.53 ± 8.51	117.15 ± 3.19
MOAC	**123.50 ±7.76**	**126.73 ± 4.33**	**124.01 ± 6.76**

policies than the three baselines.

Fig. 3 illustrates the performance of our MOAC autonomous driving agent under different user preferences in the stochastic mixed traffic flows with normal density. Here vectors are leveraged to represent user preferences in this work. The first and second elements of the user preference vector denote trade-offs for transport efficiency and energy conservation, respectively. It can be found that the electricity consumption of the ego vehicle is smaller when the preference concerning energy conservation is larger.

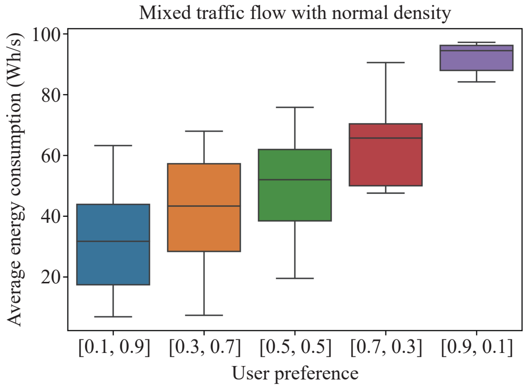


Fig. 3. The average energy consumption of our autonomous driving agent.

We implement the comparison in terms of computational cost for the MOAC and MPC methods. Under the user preference = $[0.5, 0.5]$ and the normal-density traffic flows, the computational time of the MOAC and MPC schemes are about 1.07×10^{-4} s and 3.28×10^{-2} s respectively. This is because our trained model no longer requests updating the model parameters in the test, while the MPC method needs to solve an optimization problem at each time step.

Conclusion: The results demonstrate that the MOAC autonomous driving agent can make driving decisions considering user preferences in the highway scenarios with three different traffic densities. Specifically, our agent is capable of trading off the energy and speed of AEVs according to user preferences to determine the optimal user-preference-conditioned driving policies. Additionally, the MOAC agent shows superior performance in contrast to the three baselines.

References

[1] S. Cheng, B. Yang, Z. Wang, and K. Nakano, "Spatio-temporal image representation and deep-learning-based decision framework for automated vehicles," *IEEE Trans. Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24866–24875, Dec. 2022.

[2] X. Wang, J. Sun, G. Wang, F. Allgöwer, and J. Chen, "Data-driven control of distributed event-triggered network systems," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 1, pp. 1–14, 2023.

[3] W. Liu, J. Sun, G. Wang, F. Bullo, and J. Chen, "Data-driven resilient predictive control under denial-of-service," *IEEE Trans. Automatic Control*, 2022. DOI: 10.1109/TAC.2022.3209399

[4] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[5] X. He, C. Fei, Y. Liu, K. Yang, and X. Ji, "Multi-objective longitudinal decision-making for autonomous electric vehicle: An entropy-constrained reinforcement learning approach," in *Proc. IEEE 23rd Int. Conf. Intelligent Transportation Syst.,* 2020, pp. 1–6.

[6] C. Huang, C. Lv, P. Hang, and Y. Xing, "Toward safe and personalized autonomous driving: Decision-making and motion control with DPF and CDT techniques," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 2, pp. 611–620, 2021.

[7] P. Hang, C. Huang, Z. Hu, Y. Xing, and C. Lv, "Decision making of connected automated vehicles at an unsignalized roundabout considering personalized driving behaviours," *IEEE Trans. Vehicular Technology*, vol. 70, no. 5, pp. 4051–4064, 2021.

[8] X. Xu, L. Zuo, X. Li, L. Qian, J. Ren, and Z. Sun, "A reinforcement learning approach to autonomous decision making of intelligent vehicles on highways," *IEEE Trans. Systems, Man, Cybernetics: Systems*, vol. 50, no. 10, pp. 3884–3897, 2018.

[9] C. Li and K. Czarnecki, "Urban driving with multi-objective deep reinforcement learning," in *Proc. 18th Int. Conf. Autonomous Agents MultiAgent Syst.*, 2019, pp. 359–367.

[10] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *Proc. 21st IEEE Int. Conf. Intelligent Transportation Systems,* 2018, pp. 2575–2582.

[11] S. Natarajan and P. Tadepalli, "Dynamic preferences in multi-criteria reinforcement learning," in *Proc. 22nd Int. Conf. Machine Learning*, 2005, pp. 601–608.

[12] R. Yang, X. Sun, and K. Narasimhan, "A generalized algorithm for multi-objective reinforcement learning and policy adaptation," *Advances Neural Information Proc. Syst.*, vol. 32, 2019.

[13] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. PMLR Int. Conf. Machine Learning*, 2018, pp. 1587–1596.