A Comparative Study of Deep Reinforcement Learning-based Transferable Energy Management Strategies for Hybrid Electric Vehicles

Xu, Jingyi ; Li, Z.; Gao, Li ; Ma, Junyi ; Liu, Qi ; Zhao, Yanan

DOI
10.1109/IV51971.2022.9827042

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV)

2022 IEEE Intelligent Vehicles Symposium (IV)
June 5-9, 2022. Aachen, Germany

A Comparative Study of Deep Reinforcement Learning-based Transferable Energy Management Strategies for Hybrid Electric Vehicles

Jingyi Xu[1], Zirui Li[2], Li Gao[1], Junyi Ma[1], Qi Liu[1] and Yanan Zhao[1,*]

2022 IEEE Intelligent Vehicles Symposium (IV) | 978-1-6654-8821-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IV51971.2022.9827042

***Abstract*— The deep reinforcement learning-based energy management strategies (EMS) have become a promising solution for hybrid electric vehicles (HEVs). When driving cycles are changed, the neural network will be retrained, which is a time-consuming and laborious task. A more efficient way of choosing EMS is to combine deep reinforcement learning (DRL) with transfer learning, which can transfer knowledge of one domain to the other new domain, making the network of the new domain reach convergence values quickly. Different exploration methods of DRL, including adding action space noise and parameter space noise, are compared against each other in the transfer learning process in this work. Results indicate that the network added parameter space noise is more stable and faster convergent than the others. In conclusion, the best exploration method for transferable EMS is to add noise in the parameter space, while the combination of action space noise and parameter space noise generally performs poorly. Our code is available at `https://github.com/BIT-XJY/RL-based-Transferable-EMS.git`.**

I. Introduction

Hybrid electric vehicles (HEVs) are currently important carriers of self-driving technology [1]. HEVs involve two or more energy sources. Thus there is a considerable need for energy management strategies (EMS) to distribute power supplements among several power sources to improve energy efficiency and reduce emissions [2]. There are mainly three types of EMS for HEVs: rule-based methods, optimization-based techniques, and learning-based approaches [3].

The rule-based approach is the most common method to achieve real-time control of HEVs, the effectiveness of which depends on the intuition and experience of engineers [4]. To reduce the reliance on professional engineers, the optimization-based method is introduced, using optimization algorithms to solve for optimal or sub-optimal solutions in the feasible domain to obtain better fuel economy [5]. According to different optimal control objectives and algorithms, optimization-based EMS can be divided into global optimization and real-time optimization approaches. The classical optimization-based methods include linear programming algorithm (LP) [6], dynamic programming algorithm (DP) [7], [8], equivalent fuel consumption minimization strategy (ECMS) [9], [10], model predictive control (MPC) [11], etc. The above methods improve the real-time performance and fuel economy of EMS to some extent, but they have more computational cost than rule-based methods [12].

With the rapid development of machine learning in recent years, the learning-based energy management method has become a promising solution for HEVs. Current studies mainly focus on deep reinforcement learning (DRL) based EMS due to their strong learning ability, where the EMS problem is modeled as a Markov Decision Process (MDP). The optimal solution for EMS can be learned through the interaction between agents and the environment. [13] used deep Q-learning network (DQN) algorithm for energy management, which solved the dimensional catastrophe problem. Based on this, [14] compared double deep Q-learning with DQN for energy management of plug-in hybrid vehicles and demonstrated advantages of the former in terms of convergence and fuel economy. [15] showed that the energy management policy based on deep deterministic policy gradient (DDPG) algorithm has a strong characterization capability of deep neural networks and can improve fuel economy significantly. In addition, [16] indicated that asynchronous advantage actor-critic (A3C) and distributed proximal policy optimization (DPPO) improved the learning efficiency.

Although DRL-based methods have made a significant breakthrough, their limitations are the long training time for an agent to learn the optimal solution through trial-and-error interactions with the environment [17]. Besides, the training process must be repeated even when encountering a new but similar task. Therefore, some works have combined transfer learning with DRL to improve the training efficiency between similar tasks. [18] combined proximal policy optimization (PPO) and transfer learning to effectively reduce time consumption and guarantee control performance. [19] combined DDPG and transfer learning to derive an adaptive energy management controller for hybrid tracked vehicles. Results show that this method has the potential to be applied in real-world environments. [20] incorporated transfer learning into DDPG-based EMS for HEVs to transfer knowledge among three types of HEVs that have apparently different structures.

In DRL, the agent utilizes exploration methods to acquire knowledge about the environment, which may explore better actions. The main approach is to add different types of

This work was supported by the National Key R&D Program of China under Grant 2018YFB0105205-02, Grant 2017YFC0804803 and Grant 2017YFC0804808.

[1] Jingyi Xu, Li Gao, Junyi Ma, Qi Liu and Yanan Zhao are with School of Mechanical Engineering, Beijing Institute of Technology, 100081 Beijing, China

[2] Zirui Li is with the Department of Transport and Planning, Delft University of Technology. Delft 2628 CD, The Netherlands and is also with the School of Mechanical Engineering, Beijing Institute of Technology, 100081 Beijing, China

* Corresponding author: Y. Zhao and Z. Li

noise while selecting actions. Comparing the impact of different exploration methods on DRL is implemented by much previous work [21], [22]. However, there are few studies considering the effects of exploration methods on the combination of DRL and transfer learning, which improve the training efficiency of algorithms and reduce the computational cost.

This work is a comparative study, which focuses on effects of different exploration methods of DDPG for transferable EMS. DDPG combines advantages of DQN and the actor-critic architecture, which leads to stability and high efficiency. Thus, DDPG is appropriate for evaluating the strategies from network parameters transferring. In this work, several types of noise are added to DDPG netwoks which are trained by multiple driving cycles. Then, training weights are saved to initialize a new DDPG network. The second training process is performed with noise to acquire the optimal transferable EMS.

In sum, the main contributions of this work are: Different types of noise, i.e., action space noise and parameter space noise, are added to the DDPG algorithm to explore in actions selection. Parameters of networks with different exploration methods are used to initialize new networks. The methods of exploration that work best for DDPG-based EMS and suit the most for transfer learning in the training efficiency are given by the comparative study.

The remainder of this work is organized as follows: Section II introduces the proposed method in comparing effects of different exploration approaches of DDPG-based EMS and the performance of the transferred new network; Section III details experiment results, and the conclusion is depicted in Section IV.

II. Deep Reinforcement Learning-based Transferable Energy Management Strategies

A DRL-based transferable EMS is used to evaluate the performance of different exploration methods. The sketch map of DRL-based transferable EMS is shown in Fig.1. This section describes the HEV model, the DRL-based EMS formulation, different types of noise added to DRL networks, and the effects of transferred new domain networks using different kinds of noise. In this process, the effects of different types of noise for exploration in DDPG and deep transfer learning are compared in detail in Section III.

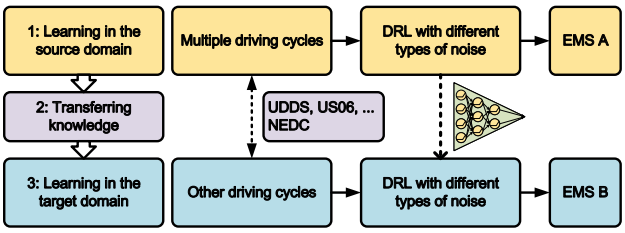


Fig. 1. The sketch map of DRL-based transferable EMS.

A. Hybrid Electric Vehicle Model

The EMS for Prius, one of the most classical HEVs, has been extensively studied [23].

1) Prius configuration: Prius is equipped with the Hybrid Synergy Drive system, which consists of an internal combustion engine ICE, an electric motor MG2, and a generator MG1. Prius is also equipped with a low-capacity nickel-metal hydride (Ni-MH) battery used to drive the motor and generator. These systems in Fig.2 are integrated with a power splitting planetary gear, which provides various power flow configurations for different operating.

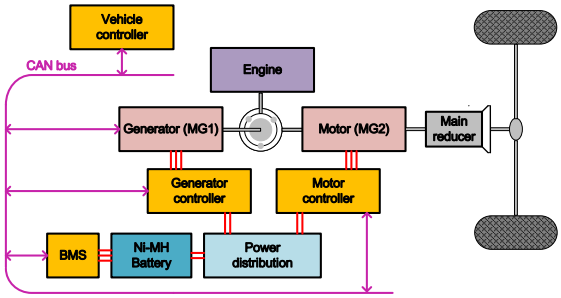


Fig. 2. Architecture of Prius powertrain.

2) Power request model: After building the Prius model, the vehicle power demand is calculated using the longitudinal force balance equation. The longitudinal force F consists of rolling resistance F_f, aerodynamic drag F_w, gradient resistance F_i, and inertial force F_a [24]:

$$\begin{cases} F = F_f + F_w + F_i + F_a \\ F_f = mg \cdot f \\ F_w = \dfrac{1}{2}\rho \cdot A_f \cdot C_D \cdot v^2 \\ F_i = mg \cdot i \\ F_a = m \cdot a \end{cases} \tag{1}$$

where m is the curb weight, g is the gravitational constant, f is the rolling friction coefficient, ρ is the air density, A_f is the fronted area, C_d is the aerodynamic coefficient, v is the speed in regard to a certain driving cycle, i is the road slope (not considered in this paper), and a is the acceleration.

3) Powertrain system model: The engine, the electric motor, and the generator of the Prius are modeled by their corresponding efficiency maps from bench tests. The Ni-MH battery is modeled by an equivalent circuit model ignoring temperature changing and battery aging:

$$\begin{cases} P(t) = I(t) \cdot V_{oc}(t) - R_0 \cdot I^2(t) \\ I(t) = \dfrac{V_{oc}(t) - \sqrt{V_{oc}^2(t) - 4 \cdot R_0 \cdot P(t)}}{2R_0} \\ SoC(t) = \dfrac{Q_0 - \int_0^t I(t)dt}{Q} \end{cases} \tag{2}$$

where P is the output power, I denotes the current, V_{oc} is the open-circuit voltage, R_0 is the internal resistance, SoC is the state of charge, Q_0 is the initial battery capacity, and Q

TABLE I
PARAMETERS OF PRIUS

Components	Parameters	Values
Engine	Maximum power, P_e	56 kW
	Maximum torque, T_e	120 Nm
Motor	Maximum power, P_m	50 kW
	Maximum torque, T_m	400 Nm
Battery	Capacity, Q	1.54 kWh
	Voltage, V_oc	237 V
Vehicle	Curb weight, m	1449 kg
	Roll resistance coefficient, f	0.013
	Air resistance coefficient, f_A	0.26
	Frontal area, A_f	2.23 m^2
	Wheel radius, r	0.287 m
Transmission	Final gear ratio, i_g	3.93
	Characteristic parameter, C	2.6

is the nominal battery capacity. Details on Prius parameters are shown in Table I.

B. DRL Formulation

A DRL problem that satisfies the Markov property can generally be modeled in terms of the MDP, which can be characterized as $(\mathbf{S},\mathbf{A},\mathbf{P},R,\gamma)$. $\mathbf{S}$ represents a set of state spaces. $\mathbf{A}$ is a set of action spaces. $\mathbf{P}$ denotes a state transition probability matrix. R represents a reward function. γ denotes a discount factor.

DDPG is one of the most typical actor-critic DRL methods, which is an off-policy and model-free algorithm. As illustrated in Fig.3, DDPG has an actor network $\mu(\mathbf{s}|\theta^{\mu})$, a critic network $Q(\mathbf{s},\mathbf{a}|\theta^{Q})$, an actor target network $\mu'(\mathbf{s}'|\theta^{\mu'})$, and a critic target network $Q'(\mathbf{s}',\mathbf{a}'|\theta^{Q'})$. The actor target network has the same structure as the actor network, while the critic target network has the same structure as the critic network. $\mathbf{s}$ is the agent state as the input of actor network and critic network. $\mathbf{a}$ is the agent action as the output of actor network and the input of critic network. θ represents parameters of the corresponding network. $\mathbf{s}'$ and $\mathbf{a}'$ are defined in the same way.

The DDPG algorithm is used to learn the optimal policy of Prius EMS in this work. The neural network has a pyramid-like architecture, with the number of neurons in hidden layers decreasing layer by layer. The optimal EMS utilizes the DDPG algorithm, which is trained with different driving cycles.

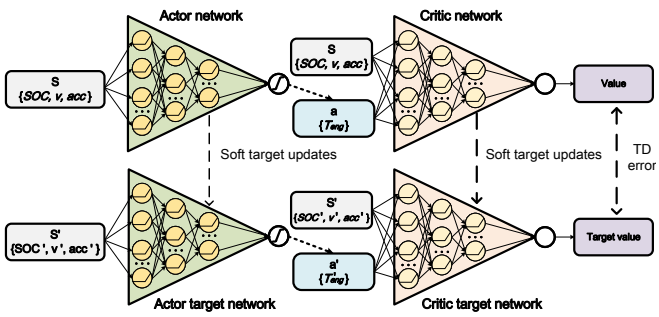


Fig. 3. The architecture of DDPG.

The DDPG-based EMS is formulated according to the following MDP.

1) State space, $\mathbf{S}$*:* The state of the system,

$$\mathbf{s} = \{SoC, v, acc\} \tag{3}$$

which consists of SoC, the velocity of Prius v and the acceleration acc.

2) Action space, $\mathbf{A}$*:* At each episode, the agent can select actions in continuous engine power T_{eng}:

$$\mathbf{a} = \{T_{eng}\} \tag{4}$$

3) Reward function, R: There are two aspects of the reward function of DDPG-based EMS, the energy consumption and the SoC sustaining. The multi-objective reward function is defined as:

$$r = -\{\alpha[fuel(t) + elec(t)] + \beta[SoC_{ref} - SoC(t)]^n\} \tag{5}$$

where α is the weight of Prius consumption including the fuel consumption of engine $fuel$ and the electricity consumption of motor $elec$, β is the weight of battery charge-sustaining, and SoC_{ref} represents the SoC reference value. The goal of the reward function is to minimize the energy consumption and retain the battery SoC at an appropriate range. To control for variables, in the following comparison, SoC_{ref} is selected as 0.6 according to the minimum charge-discharge internal resistance. α is selected as 1, β is set to 350, and n is set to 2, according to the previous work [24].

C. Transfer Learning

Traditional DRL algorithms are used to solve the problem with training and test data in the same domain. However, once the domain is changed, the network needs to be retrained, which is quite complex and time-consuming [25], [26], [27]. Transfer learning is extremely useful in solving this problem. When two domains are similar, network parameters can be stored and reused in the new one along with transfer learning approaches [28], [29], [30].

Given a source domain M_s and a target domain M_t, transfer learning aims to learn an optimal policy π^* from M_s for M_t. M_s provides prior knowledge D_s that is accessible for M_t. Thus, by leveraging the information from D_s, the target agent learns better and faster in M_t [31].

A network that specializes in obtaining source EMS is used in our work. Since driving cycles of M_s and M_t have the same feature space and are correlated with each other, source domain knowledge can be transferred to the novel, but relevant target domain [32]. The majority of parameters in the neural network are the same, and only parameters of the output layer should be retrained. Thus, both the source network and the target network use the same DDPG architecture shown in Fig.3, and the weights of the source network except for the last layer are used to initialize the target network that will be trained on new driving cycles. Further details about hyperparameters of DDPG in M_s and M_t are given in Table II.

TABLE II
DDPG HYPERPARAMETERS

Parameters	Source Domain	Target Domain
Number of Episodes, K	1000	300
Replay memory size, M	50000	50000
Learning rate of actor network, lr_a	0.001	0.0009
Learning rate of critic network, lr_c	0.01	0.009
Discount factor, γ	0.9	0.9
Target network update frequency, τ	0.01	0.01
Mini-batch size, *batch*	64	64

TABLE III
NETWORKS ADDED DIFFERENT TYPES OF NOISE

Space Added Noise	Noise Type	Variance
Action space	Gaussian	0.02,0.03,0.04,0.05,**0.06**
	OU	0.08,**0.09**,0.10,0.11,0.13
Parameter space	Gaussian	**0.03**,0.04
Action & parameter space	Gaussian & Gaussian	0.06 & 0.03
	OU & Gaussian	0.09 & 0.03

In DDPG, the agent utilizes exploration to acquire knowledge about the environment and applies exploitation to select a control action based on current knowledge [33]. Thus, the coordination between exploitation and exploration is essential. The following parts of this subsection provide a description of exploration methods used in the DDPG-based transferable EMS.

The primary purpose of exploration is to avoid local optimum for agent's behaviors [21]. Thus, to realize efficient and effective exploration, random noise is frequently added to perturb selected actions, which is mainly focused on in our work. Main approaches can be classified into two groups: adding noise in the action space and adding noise directly to agent's parameters.

1) Action space noise: When the agent selects actions using the actor network, the noise $\mathcal{N}$ is added to the action space. The final selected action a_t at each step satisfies:

$$a_t = \mu(s|\theta^\mu) + \mathcal{N} \tag{6}$$

Action space noise could be a simple Gaussian noise or a more advanced Ornstein-Uhlenbeck (OU) correlated noise process [22]. Gaussian noise satisfies $\mathcal{N} \sim N(0, \sigma^2 I)$, where σ^2 denotes variance and the expected value is set to 0. An OU process [34] can be used as a temporally correlated noise. Just like the Gaussian noise mentioned above, the expected value of OU noise $\mathcal{N} \sim OU(0, \sigma^2)$ is set to 0, and the variance can be set to multiple values.

2) Parameter space noise: While adding noise in the action space to explore, there is no guarantee that the same action will be chosen in the same state each time, which can lead to inconsistent exploration. The parameter space noise solves this problem and directly perturbs actor network parameters to get a rich set of behaviors. The final selected action a_t at each step satisfies:

$$\begin{cases} a_t = \mu(s|\tilde{\theta}^\mu) \\ \tilde{\theta} = \theta + N(0, \sigma^2 I) \end{cases} \tag{7}$$

III. EXPERIMENTS

The purpose of our work is to compare effects of different exploration methods of DDPG-based EMS and transferred new networks in terms of transfer efficiency. Driving cycles are selected for the source domain and the target domain, which are different but similar. Then, networks with different exploration methods are trained in the source domain, of which parameters are saved. Finally, the adaptation of target domain networks, of which parameters are initialized using saved weights, is evaluated.

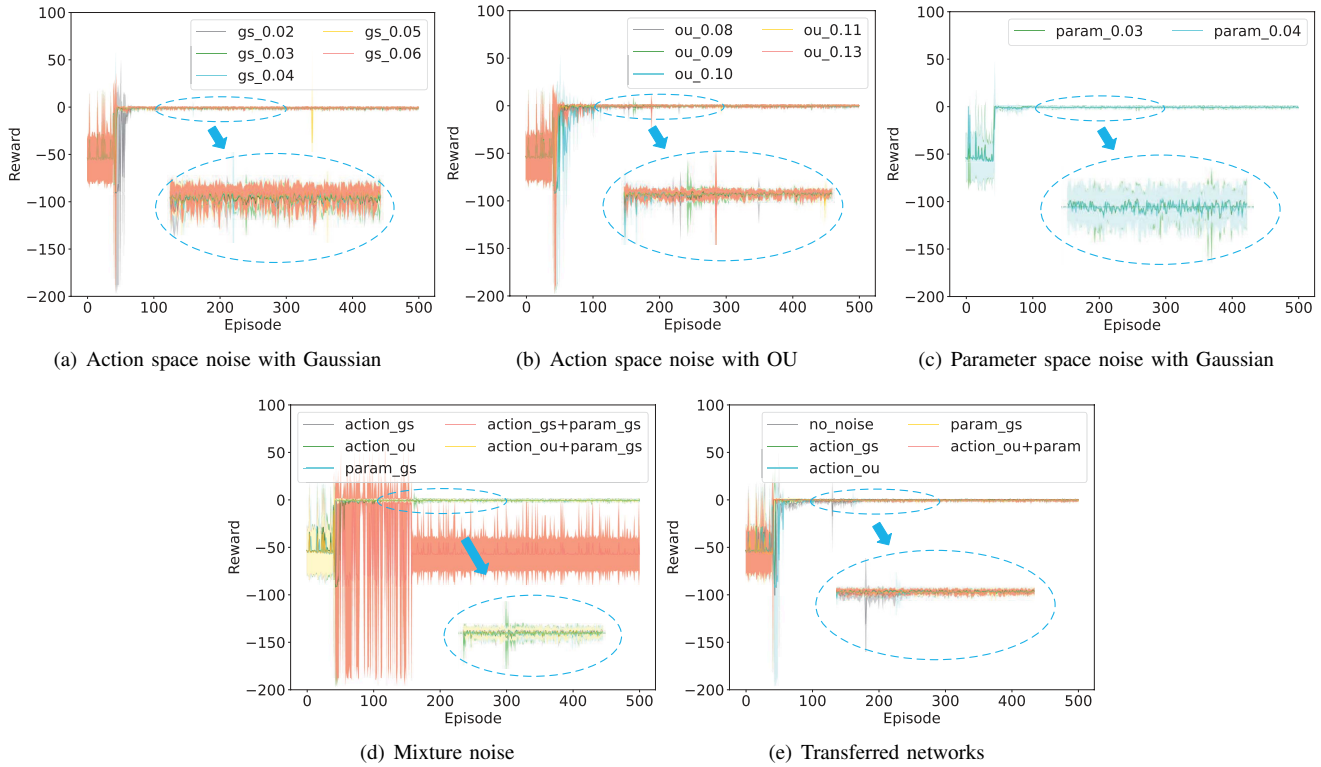
A. Driving Cycles

In this work, driving cycles are all selected from standard data. Source tasks are performed over multiple cycles, including Urban Dynamometer Driving Schedule (UDDS) [35], FTP75 [24], etc. Target tasks are conducted on New European Driving Cycle (NEDC) [36], which is different from driving cycles used in the source domain but similar. Using multiple driving cycles for training in the source domain improves the generalization ability of the trained model, which leads to better transfer results. A driving cycle with a high similarity to the source driving cycles is chosen for the target domain, since similarity is a necessary factor for transfer learning.

B. Training in the Source Domain

To ensure the validity of weights to be transferred, networks with different exploration methods are firstly trained on the source domain. Settings of networks with different types of noise are shown in Table III. By comparative studying, suitable networks, of which parameters are used to initiate weights of target networks, are selected according to the training results.

As described in Fig.4, Gaussian noise added in the action space, OU noise added in the action space, Gaussian noise added in the parameter space, and their mixture are used to explore in DDPG-based EMS. In Fig.4(a), different variance values σ^2 of Gaussian noise added in the action space are set to 0.02, 0.03, 0.04, 0.05, 0.06, respectively. The reward fluctuates the most when the variance is set to 0.02. Only the network with variance 0.05 shows considerable oscillations once the trained weights converge. Thus, the network with variance 0.06 is the most stable. Similarly, Fig.4(b) and 4(c) illustrate that the network's variance of 0.09 with OU noise added in the action space and the network's variance of 0.03 with Gaussian noise added in the parameter space are the most stable, respectively. Results of other noise configurations, which converge to a local optimum or fluctuate too much, are not shown here. Besides, different effects of a single noise and a mixed noise are compared in Fig.4(d). Results using the Gaussian noise added in the action space and parameter space are very unstable, as shown by the yellow line. The noise of multiple Gaussian processes makes the agent tend to explore rather than exploit to a great extent, leading to more non-optimal actions with fluctuating



(a) Action space noise with Gaussian

(b) Action space noise with OU

(c) Parameter space noise with Gaussian

(d) Mixture noise

(e) Transferred networks

Fig. 4. Comparison of different exploration methods in the source domain.

reward values. The most stable is the Gaussian noise added in the action space, followed by the Gaussian noise added in both the action space and parameter space.

Above all, the most stable network is the one that utilizes the Gaussian noise added in the action space with 0.06 variance to explore, followed by the network with the combination of OU noise added in the action space and Gaussian noise added in the parameter space to explore. Chosen networks which are transferred to a new domain are shown in Fig.4(e).

C. Adaptation of Transfer Learning

Results of the transferred DDPG in the target domain using different exploration methods in EMS are discussed in this subsection. A new network is trained on the new driving cycle, learning from scratch or initializing the network parameters using prior ones.

As shown in Table IV, following criteria are adopted to evaluate the adaptation of the transferable EMS [37]:

1) Jumpstart Performance (JP): The initial performance of the agent. The mean reward of the first 50 episodes is used to evaluate it.

2) Asymptotic Performance (AP): The ultimate performance of the agent. The mean reward of the convergence interval is used to evaluate it.

3) Time to Threshold (TT): The learning time needed for the target agent to reach a certain performance threshold. The iteration number of convergence is used to evaluate it.

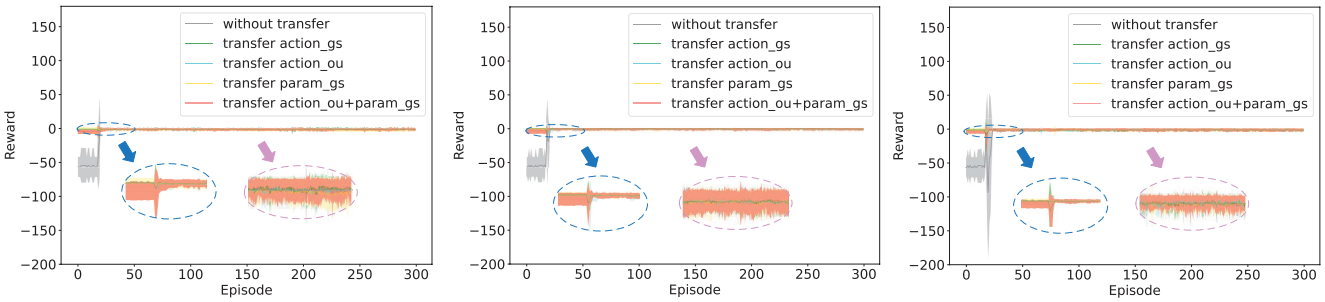
In Table IV, TFS means the source network trained from scratch. Gaussian_AS, OU_AS, Gaussian_PS and APS mean source networks with Gaussian noise added in the action space, OU noise added in the action space, Gaussian noise added in the parameter space, and both OU noise added in the action space and Gaussian noise added in the parameter space, respectively. While target networks use the same type of noise, the maximum reward value obtained from first 50 episodes is larger than the maximum convergence value in the convergence interval, which means that the DDPG network falls into a local optimum during the exploration process. Besides, the initialization of parameters of target networks with parameter space noise generally works well in terms of convergence speed and mean reward value, except for target networks using the mixture of action space and parameter space noise for exploration. After target networks have converged, convergence values of different exploration methods do not differ significantly.

As shown in Fig.5(a), no noise is added on the target network. In first 50 episodes, the network which is trained from scratch (gray line) starts out with a small reward value, which means that its JP is poor. The green, blue, yellow, and orange lines represent parameters of networks with different exploration methods to initialize current target networks. Weights of these networks are perturbed by Gaussian noise added in the action space ($\sigma^2 = 0.06$), OU noise added in the action space ($\sigma^2 = 0.09$), Gaussian noise added in the parameter space ($\sigma^2 = 0.03$), both OU noise added in the action space noise and Gaussian noise added in the parameter space to explore, respectively. The yellow line fluctuates the least. In the convergence interval (50 300 episodes in all

TABLE IV

MEAN REWARD AND ITERATION NUMBER OF TARGET NETWORK

Exploration Method (Target Network)	Transferred Network Parameter (Source Network)	Mean Return (First 50 Episodes)	Mean Return (Convergence Interval)	Iteration Number
No noise	TFS	-21.6386	-1.2101	30
	Gaussian_AS	-0.9186	**-1.0540**	20
	OU_AS	-0.9451	-1.2019	29
	Gaussian_PS	**-0.9009**	-1.4133	**21**
	APS	-2.0715	-1.1878	29
Gaussian noise added in the ation space	TFS	-20.9453	**-0.8766**	36
	Gaussian_AS	-0.8780	-0.8776	35
	OU_AS	-0.9748	-0.8908	30
	Gaussian_PS	**-0.7987**	-0.9071	**25**
	APS	-1.7628	-0.9159	32
OU noise added in the ation space	TFS	-25.3330	-1.1527	36
	Gaussian_AS	-1.1239	-1.0585	24
	OU_AS	-0.9035	**-1.0417**	**22**
	Gaussian_PS	**-0.8720**	-1.1611	31
	APS	-2.3262	-1.0474	23
Gaussian noise added in the parameter space	TFS	-21.6009	-1.1523	35
	Gaussian_AS	**-1.0250**	**-1.1132**	**21**
	OU_AS	-1.1109	-1.1652	22
	Gaussian_PS	-1.5669	-1.2466	34
	APS	-4.4550	-1.2734	30
Noise added in ation space and parameter space	TFS	-21.1288	-1.0919	35
	Gaussian_AS	-1.4819	-1.0066	31
	OU_AS	**-0.9408**	-1.1473	27
	Gaussian_PS	-13.8613	**-0.9998**	**24**
	APS	-2.6120	-1.1589	30



(a) No noise (b) Gaussian action space noise (c) OU action space noise

(d) Parameter space noise (e) Action space and parameter space noise

Fig. 5. Comparison of different exploration methods in the target domain.

target cases), the green line is the most stable, gray and blue lines are the second most unstable, and the most unstable one is the yellow line. In general, they do not differ much. This means that it is the most stable to initialize the target network with parameters of the original network with Gaussian action space noise.

The Gaussian noise is added in the action space of target networks, of which the training process is shown in Fig.5(b). In first 50 episodes, the gray line fluctuates the most, and the orange line fluctuates the least, while the blue line fluctuates the least in the convergence interval. This indicates that the most stable approach is to initialize the target network with parameters of the original network with OU action space noise, and use Gaussian action space noise to explore.

The action space noise with OU process is added on the target network, of which the training process is shown in Fig.5(c). The gray line has the most considerable fluctuation, followed by the orange line in first 50 episodes. The blue line has the slightest fluctuation. In the convergence interval, the most stable one is still the blue line, which indicates that it has a better learning effect to initialize the target network, which uses OU action space noise to explore, with parameters of the original network with OU action space noise.

The Gaussian noise added in the parameter space is added on the target network, of which the training process is shown in Fig.5(d). In first 50 episodes, the gray line has a small initial value because it does not have any prior knowledge. The orange line fluctuates the second most, which means that parameters of the network with action space noise and parameter space noise are not suitable for initializing the new target network with parameter space noise. In the convergence interval, the most stable one is still the blue line, followed by the green line, and the most unstable one is the yellow line. The blue line is the most stable one throughout the training process.

Both the action space noise and the parameter space noise are added on the target network, of which the training process is shown in Fig.5(e). The gray line has the smallest initial value, while the yellow line has the largest fluctuation in first 50 episodes. It means that parameters of the network with parameter space noise are not suitable for initializing the target network with action space noise and parameter space noise. In the convergence interval, the fluctuation of each method is negligible.

Combining results expressed in Fig.4 and Fig.5, the network with the action space noise works best for DDPG-based EMS. For transfer learning, the network with the parameter space noise is the most stable, while the network with multiple noises of action space and parameter space has poor initial performance.

IV. Conclusion

In this paper, choosing an optimal and efficient EMS is formulated as a deep reinforcement learning-based transfer learning problem. We compared different exploration approaches for deep reinforcement learning and transfer learning to find out the best energy management strategy. Effects of action space noise and parameter space noise which are added to the DDPG algorithm, are presented. Experimental results show that the method of parameter space noise exploration works best for DDPG-based transferable EMS.

Despite these encouraging comparison results, there are several avenues for future research. First, we want to investigate other exploration modalities, not only adding noise for action selecting. We furthermore plan to compare different exploration methods based on more robust DRL algorithms, such as twin delayed DDPG and soft actor-critic.

References

[1] T. Liu, W. Tan, X. Tang, J. Zhang, Y. Xing, and D. Cao, "Driving conditions-driven energy management strategies for hybrid electric vehicles: A review," *Renewable and Sustainable Energy Reviews*, vol. 151, p. 111521, 2021.

[2] T. Liu and X. Hu, "A bi-level control for energy efficiency improvement of a hybrid tracked vehicle," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1616–1625, 2018.

[3] X. Guo, T. Liu, B. Tang, X. Tang, J. Zhang, W. Tan, and S. Jin, "Transfer deep reinforcement learning-enabled energy management strategy for hybrid tracked vehicle," *IEEE Access*, vol. 8, pp. 165 837–165 848, 2020.

[4] H. Guo, X. Wang, and L. Li, "State-of-charge-constraint-based energy management strategy of plug-in hybrid electric vehicle with bus route," *Energy Conversion and Management*, vol. 199, p. 111972, 2019.

[5] T. Hofman, M. Steinbuch, R. Van Druten, and A. Serrarens, "Rule-based energy management strategies for hybrid vehicles," *International Journal of Electric and Hybrid Vehicles*, vol. 1, no. 1, pp. 71–94, 2007.

[6] F. Z. Kadda, S. Zouggar, and M. L. Elhafyani, "Optimal energy management of an autonomous hybrid system by using the linear programming method," in *2013 International Renewable and Sustainable Energy Conference (IRSEC)*. IEEE, 2013, pp. 420–425.

[7] C.-C. Lin, H. Peng, J. W. Grizzle, and J.-M. Kang, "Power management strategy for a parallel hybrid electric truck," *IEEE transactions on control systems technology*, vol. 11, no. 6, pp. 839–849, 2003.

[8] X. Wang, H. He, F. Sun, and J. Zhang, "Application study on the dynamic programming algorithm for energy management of plug-in hybrid electric vehicles," *Energies*, vol. 8, no. 4, pp. 3225–3244, 2015.

[9] A. Rezaei, J. B. Burl, and B. Zhou, "Estimation of the ecms equivalent factor bounds for hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2198–2205, 2017.

[10] Q. Jiang, F. Ossart, and C. Marchand, "Comparative study of real-time hev energy management strategies," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10 875–10 888, 2017.

[11] H. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, I. V. Kolmanovsky, and S. Di Cairano, "Mpc-based energy management of a power-split hybrid electric vehicle," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 593–603, 2011.

[12] R. Johri and Z. Filipi, "Optimal energy management of a series hybrid vehicle with combined fuel economy and low-emission objectives," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 228, no. 12, pp. 1424–1439, 2014.

[13] J. Wu, H. He, J. Peng, Y. Li, and Z. Li, "Continuous reinforcement learning of energy management with deep q network for a power split hybrid electric bus," *Applied energy*, vol. 222, pp. 799–811, 2018.

[14] X. Qi, Y. Luo, G. Wu, K. Boriboonsomsin, and M. Barth, "Deep reinforcement learning enabled self-learning control for energy efficient driving," *Transportation Research Part C: Emerging Technologies*, vol. 99, pp. 67–81, 2019.

[15] Y. Wu, H. Tan, J. Peng, H. Zhang, and H. He, "Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series-parallel plug-in hybrid electric bus," *Applied energy*, vol. 247, pp. 454–466, 2019.

[16] X. Tang, J. Chen, T. Liu, Y. Qin, and D. Cao, "Distributed deep reinforcement learning-based energy and emission management strategy for hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 9922–9934, 2021.

[17] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[18] T. Liu, B. Wang, W. Tan, S. Lu, and Y. Yang, "Data-driven transferred energy management strategy for hybrid electric vehicles via deep reinforcement learning," *arXiv preprint arXiv:2009.03289*, 2020.

[19] X. Guo, T. Liu, B. Tang, X. Tang, J. Zhang, W. Tan, and S. Jin, "Transfer deep reinforcement learning-enabled energy management strategy for hybrid tracked vehicle," *IEEE Access*, vol. 8, pp. 165 837–165 848, 2020.

[20] R. Lian, H. Tan, J. Peng, Q. Li, and Y. Wu, "Cross-type transfer for deep reinforcement learning based hybrid electric vehicle energy management," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8367–8380, 2020.

[21] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," *arXiv preprint arXiv:1706.01905*, 2017.

[22] C. Colas, O. Sigaud, and P.-Y. Oudeyer, "Gep-pg: Decoupling exploration and exploitation in deep reinforcement learning algorithms," in *International conference on machine learning*. PMLR, 2018, pp. 1039–1048.

[23] D. V. Prokhorov, "Toyota prius hev neurocontrol and diagnostics," *Neural Networks*, vol. 21, no. 2-3, pp. 458–465, 2008.

[24] R. Lian, J. Peng, Y. Wu, H. Tan, and H. Zhang, "Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle," *Energy*, vol. 197, p. 117297, 2020.

[25] C. Lu, F. Hu, D. Cao, J. Gong, Y. Xing, and Z. Li, "Virtual-to-real knowledge transfer for driving behavior recognition: Framework and a case study," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6391–6402, 2019.

[26] Z. Li, C. Gong, C. Lu, J. Gong, J. Lu, Y. Xu, and F. Hu, "Transferable driver behavior learning via distribution adaption in the lane change scenario," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 193–200.

[27] C. Gong, Z. Li, C. Lu, J. Gong, and F. Hu, "A comparative study on transferable driver behavior learning methods in the lane-changing scenario," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3999–4005.

[28] Z. Li, J. Gong, C. Lu, and J. Li, "Personalized driver braking behavior modelling in the car-following scenario: An importance weight-based transfer learning approach," *IEEE Transactions on Industrial Electronics*, 2022.

[29] Z. Li, J. Gong, C. Lu, and J. Xi, "Importance weighted gaussian process regression for transferable driver behaviour learning in the lane change scenario," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 497–12 509, 2020.

[30] C. Lu, F. Hu, D. Cao, J. Gong, Y. Xing, and Z. Li, "Transfer learning for driver model adaptation in lane-changing scenarios using manifold alignment," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 8, pp. 3281–3293, 2019.

[31] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *arXiv preprint arXiv:2009.07888*, 2020.

[32] X. Guo, T. Liu, B. Tang, X. Tang, J. Zhang, W. Tan, and S. Jin, "Transfer deep reinforcement learning-enabled energy management strategy for hybrid tracked vehicle," *IEEE Access*, vol. 8, pp. 165 837–165 848, 2020.

[33] X. Hu, T. Liu, X. Qi, and M. Barth, "Reinforcement learning for hybrid and plug-in hybrid electric vehicle energy management: Recent advances and prospects," *IEEE Industrial Electronics Magazine*, vol. 13, no. 3, pp. 16–25, 2019.

[34] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical review*, vol. 36, no. 5, p. 823, 1930.

[35] N. Yang, L. Han, C. Xiang, H. Liu, and X. Hou, "Energy management for a hybrid electric vehicle based on blended reinforcement learning with backward focusing and prioritized sweeping," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3136–3148, 2021.

[36] T. Liu, X. Hu, W. Hu, and Y. Zou, "A heuristic planning reinforcement learning-based energy management for power-split plug-in hybrid electric vehicles," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6436–6445, 2019.

[37] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *arXiv preprint arXiv:2009.07888*, 2020.