DATA-EFFICIENT DEEP REINFORCEMENT LEARNING WITH CONVOLUTION-BASED STATE ENCODER NETWORKS

Qiang Fang,* Xin Xu,* Yixin Lan,* Yichuan Zhang,* Yujun Zeng,* and Tao Tang*

Abstract

Due to its ability to deal with high-dimensional end-to-end learning control problems, deep reinforcement learning (DRL) has received lots of research interests in recent years. However, the existing DRL approaches still face the challenge of data efficiency and the online learning control performance of DRL algorithms still needs to be improved. In this paper, we propose an online DRL approach with convolutional encoder networks. In the proposed approach, a cascaded learning control architecture is designed, which performs system state extraction and dimension reduction in the first stage and executes online reinforcement learning in the second stage. A convolutional network is used to encode features from the raw image data so that the algorithm can be implemented based on the encoded low-dimensional features, which can significantly improve the learning efficiency. Experimental results on two benchmark of learning control tasks show that the proposed approach outperforms previous end-to-end DRL approaches, which demonstrates the effectiveness and efficiency of the proposed approach.

Key Words

Deep reinforcement learning, actor-critic learning, learning control, online learning, auto-encoder

1. Introduction

As is well known, the sequential decision-making tasks are usually formulated as Markov decision processes (MDPs) [1, 2] which can be solved using dynamic programming methods when the dynamics models are known as a priori. However, traditional dynamic programming methods can only be used to solve MDPs with complete model information and discrete state or action spaces. To deal with task uncertainty, reinforcement learning (RL) algorithms have been widely studied due to their ability to learn an optimal or near-optimal policy by interacting with the environment, and various RL algorithms have been developed [3, 4, 5] in the past decades.

Nowadays, there are many real-world tasks with high-dimensional inputs or continuous spaces, and traditional RL methods might not be suitable for such complex problems. Due to the ability of solving feature extraction and approximation problems with high-dimensional inputs, such as images, deep learning has been widely studied in recent years and many successful applications of deep learning have been achieved. Subsequently, deep reinforcement learning (DRL) [6] which combines deep learning with RL was studied to solve the sequential decision-making tasks with high-dimensional inputs.

Despite the advances that have been made in the past decade, there are still some challenges for DRL in many real-world applications. One is how to improve the online data efficiency of DRL algorithms, especially when using images as input. To improve the learning efficiency of RL, a state encoder networks-based DRL (SEN-DRL) approach is presented where hierarchical encoder networks are designed to encode the input images into lower-dimensional states. Based on the encoded states, a cascaded learning control method is designed, where feature extraction and dimension reduction are performed firstly and then the online RL algorithms with encoded features are designed for MDPs with discrete actions and continuous actions, respectively. The hierarchical encoding network is based on convolutional neural networks (CNNs) [7] which is trained based on stochastic gradient descent. As raw images are encoded into low-dimensional features, efficient online RL algorithms can be designed using the encoded features, which can significantly improve the data efficiency of the learning systems.

The main contributions of this paper are summarized as follows:

- A SEN-DRL approach is presented which performs state extraction and dimension reduction using a convolution-based encoding network at first, and then executes online learning control. The proposed approach can realize fast online end-to-end learning by mapping the raw image data into lower-dimensional states to improve learning efficiency.

* College of Intelligence Science and Technology, National University of Defense Technology, Changsha, 410073, P. R. China; e-mail: {qiangfang, xinxu}@nudt.edu.c
Corresponding authors: Qiang Fang, Xin Xu
Recommended by Prof. Xinyu Wu
(DOI: 10.2316/J.2021.206-0763)

- The efficiency and effectiveness of the proposed approach are demonstrated in the learning control tasks with discrete actions and continuous actions, respectively. Simulation results on two benchmark learning control tasks show that the proposed approach outperforms pervious end-to-end DRL approaches.

The remainder of this paper is summarized as follows: we firstly introduce the related work in Section 2. In Section 3, our approach based on cascaded encoding networks is presented. In Section 4, experiments on two learning control benchmarks are conducted to show the effectiveness of our proposed method. Finally, a conclusion is drawn in Section 5.

2. Related Work

Three categories of RL algorithms with function approximation have been popularly studied in the literatures [8, 9, 10, 11, 12, 13, 14, 15, 16]. The first category of approximate RL methods is to approximate the value functions of MDPs and generate control policies by searching greedy actions. This category of RL methods includes Q-learning with neural networks, state–action–reward–state–action (SARSA) learning with function approximation, etc. The second category directly uses a function approximator to estimate the policy function of an MDPs and a near-optimal or sub-optimal control policy can be learned using policy gradient or other policy search methods. The third category uses an actor-critic learning control structure which can be viewed as a combination of value function approximation and policy function learning. Many recently developed methods belong to actor-critic-based learning control methods. In addition, actor-critic methods have been successfully applied in various domains such as robot control [17], power systems [18], cognitive radio networks [19], and finance [20]. Although human experience may be used in the selection process of approximation structures or basis functions, it is critical for an RL agent to learn appropriate features or representation for MDPs in a data-driven way. This problem is also called representation learning in the machine learning community [21, 22]. In the past decades, there have been various progresses in representation learning for RL in MDPs with large or continuous state space. In [23, 24, 25, 26], kernel methods as well as manifold learning-based methods have been studied for value function approximation in RL. However, it is still challenging to deal with learning control problems with high-dimensional inputs such as raw image data.

As a class of representation learning methods with deep network architecture, deep learning has recently been popularly studied and various state-of-the-art results have been obtained in pattern recognition and computer vision domains [7, 27, 28, 29, 30, 31, 32, 33]. Similar to the hierarchical structure of human vision systems, deep learning models can learn multi-level features from raw image data both in an unsupervised manner and with supervised signals. To deal with high-dimensional state space in sequential decision-making problems, deep reinforcement learning (DRL) has received increasing research interest in recent years.

To improve the performance of DRL algorithms, Mnih [6] proposed a deep Q-network (DQN) algorithm to deal with the control problem on Atari games from the pixels, which treats the neural networks as function approximators. However, DQN may result in overoptimistic value estimates due to the inaccuracies or noises in value function approximation. To solve such issue, Hasselt [34] proposed a double estimator method (DDQN), separating the selection of the estimator and its relevant value. Based on the deterministic policy gradient theorem, the deep deterministic policy gradient [35] algorithm considers the output of the policy as the direct representation, which allows such algorithm and its variants to deal with continuous actions. To improve the stability of policy updates, a constraint-based method called TRPO [36] was introduced. It adopts the KL divergence to improve the policy update process. Instead of KL constraints, proximal policy optimization (PPO [37]), a variant of TRPO, considers the constraint as an objective function of penalty or clipping. The latter uses model-based strategies to accelerate the learning procedure. Hester [38] proposed a model-based RL algorithm with intrinsic motivation. A prediction network (VPN) was proposed to integrate model-free and model-based RL into one neural network [39]. Moreover, some research focused on developing software or hardware implementations for improving the computational efficiency of DRL algorithms, such as the asynchronous advantage actor-critic [40] algorithm, which was designed to make use of multi-threading of GPU.

Despite the above advances in DRL, it is still necessary to improve the data efficiency of online RL algorithms with high-dimensional inputs. To directly use the original image to realize the navigation of mobile robots, Tai *et al.* [41] constructed a feature representation model based on CNN, which was used for DQN training after completing pre-training, and finally achieved good results. However, the training of CNN needs a prior control policy. In this paper, we will consider the pre-training of CNN without a prior control policy. With the rapid development of deep learning and RL methods, Szoke [42] used the CNN model with long-short-term memory to represent the state features in the highway scene image, and a driving policy to realize self-driving on the highway using the policy gradient RL algorithm. In addition, Bae *et al.* [43] proposed a multi-robot path planning algorithm using deep Q learning combined with convolution neural network to analyze the exact situation using image information. However, the training efficiency of the above-mentioned RL method combined with feature coding needs to be improved, and in this paper we propose a cascading method where the encoding process can be processed offline, thereby improving the overall training efficiency.

3. Proposed Approach

Figure 1 shows the whole framework of our approach. Here, we consider the high-dimensional images as inputs and an encoding network based on CNNs is utilized to encode states from the raw image firstly, then an RL method is chosen according to different tasks, such as discrete actions

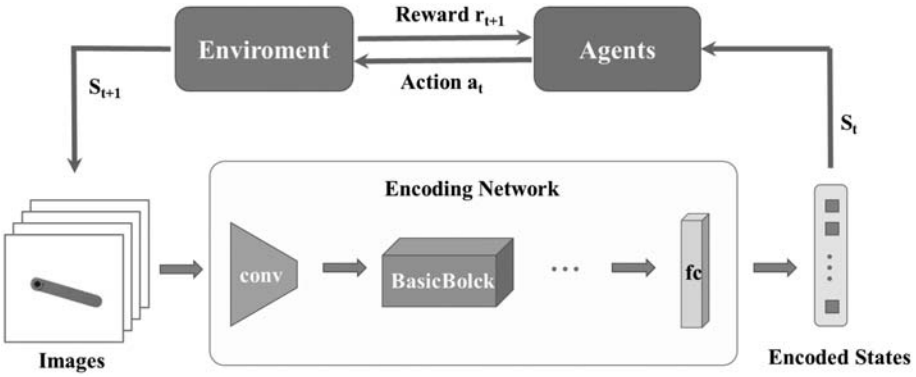


Figure 1. Our proposed approach (SEN-DRL). Different from the traditional RL approach, the raw image is considered as the input. The training of the encoding module is independent of the online learning control module and it can be done offline, which can additionally improve the sample efficiency during the whole online learning process.

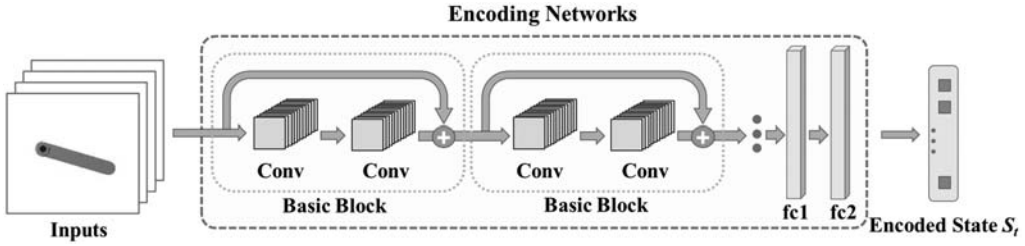


Figure 2. State encoding network.

or continuous actions. Note that our approach uses the raw image as the input, but the training of the encoding module is independent of the online learning control module and it can be done offline, which can additionally improve the sample efficiency during the whole online learning process.

In the state encoding stage, the encoding module is trained by supervised learning to encode high-dimensional observation to low-dimensional state. Figure 2 shows the details of the network. The encoding module contains four basic blocks and two fully connected layers. In each basic block, there are two convolution layers and one shortcut to jump over the two convolution layers. Shortcuts not only effectively avoid the degradation of deep networks but also can reduce the impact of vanishing gradients, as there are fewer layers to propagate through. Two fully connected layers transform the dimension of final convolution layers into the dimension determined by the physical state of the task. The training data are the task-related observation-state pairs. Moreover, it can be plugged into any RL algorithm to encode state from high-dimensional images and speed up the learning process. In the second stage, as the system states are encoded by the network, we can use RL algorithms to solve the learning control problem. As we know, the learning control problem in RL can be modeled as an MDP $\{S, A, R, P\}$, where $S \subseteq R^n$ and $A \subseteq R$ are the state space and action space, respectively. The reward function is denoted by R. The state transition probability P is determined by the system dynamics which is usually unknown or partially known for the learning controller.

According to the Bellman's optimality principle, the optimal cost at time t is equal to

$$J^*(s(t)) = \min_{u(t)} \{r(s(t), u(t)) + \gamma J^*(s(t+1))\} \quad (1)$$

The optimal control at time t is $u^*(t)$, which satisfies

$$u^*(t) = arg \min_{u(t)} \{J^*(s(t))\} \quad (2)$$

The critic learning is to approximate the value functions. Denote s_t, r_t, u_t as the abbreviations of $s(t)$, $r(s(t),u(t))$, and $u(t)$, respectively. Based on the Bellman equation for stationary control policies, the following equation can be derived:

$$V^\pi(s_t) = E^\pi[r_t + \gamma V^\pi(s_{t+1})] \quad (3)$$

where $E^\pi[\cdot]$ is with respect to the state transition probability by following a stationary action policy π.

To deal with MDPs with discrete (low-dimensional) state or action spaces, many traditional RL methods have been proposed and SARSA [1] is a popular algorithm. SARSA algorithm is an on-policy algorithm and it is a typical value-based RL algorithm. The details of the SARSA algorithm can be seen in [1, 2]. It updates the Q function based on the action executed by the current policy, considering one update process of Q function. Then the Q function can be updated based on the following equation:

$$Q(s_t, a_t) \longleftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (4)$$

where α and γ are the learning rate and the discount factor, respectively. s_t is the current state, $r_t + 1$ and $s_t + 1$ are the reward and state after executing the action a_t. With discrete action space, we combine the encoding network and SARSA to deal with the control problem in this paper and the details of our approach (SEN-DRL) are listed in

Algorithm 1. To deal with MDPs with large (continuous) state or action space, although the value-based and the policy-based methods have been investigated to improve the ability of generalization for RL algorithms, both of them have some limitations. The actor-critic learning control framework can be viewed as a combination of value function approximation and policy search. In the actor-critic learning controller, a critic network is designed to learn the value functions of the underlying MDPs and an actor network is used to learn near-optimal policies using the estimated value function in the critic to perform policy searching. Policy gradient methods often play as the actor role in an actor-critic controller, which works by computing an estimator of the policy gradient and plugging it into a gradient ascent algorithm. The most frequently [1][2] used policy gradient form in an actor is as follows:

$$\bigtriangledown J(\theta) = E_t \left[\bigtriangledown_\theta log\pi \left(a_t | s_t \right) A \left(a_t, s_t \right) \right] \tag{5}$$

where $E_t[.]$ indicates the empirical average over a finite batch of samples, and advantage function $A(a_t, s_t) = Q(a_t, s_t) - V(s_t)$. With sufficient samples, the expectation of $A(a_t, s_t)$ can be approximated accurately. However, it is impractical to sample too much so that the variance of $A(a_t, s_t)$ is large. Therefore, a value-based critic is estimated by the temporal difference (TD) [1] algorithm.

Algorithm 1 SEN-DRL for discrete actions

1: **Input:** a trained deep encoder network, a stop criterion.
2: **Output:** Q-value table.
3: **Initialization:**
I: perceived image that capture the MDPs state;
f_l: the learning factor of the actor neural network;
t: the step in each learning episode and set $t = 0$.
4: **repeat**
5: Use the trained deep encoder network to encode the current perceived image I_t and utilize the encoded result s_t to calculate the output of the actor neural network y_t;
6: Choose the actual action a_t according to policy derived from the Q-value table($\epsilon - greedy$);
7: Apply a_t to the MDPs and perceive the state transition (s_t, s_{t+1}) and compute reward $r_t = r\left(s_t, s_{t+1}\right)$;
8: Update the Q-value for the state;
9: Set $t = t + 1$
10: **until** the stop criterion is met.
11: RETURN Q-value table.

Recently, some new actor-critic algorithms have been proposed, such as the PPO algorithm [2]. The PPO algorithm has been shown to have good convergence property and it is simple to be implemented and tuned. PPO has become the default RL algorithm at OpenAI for its ease of use and good performance. PPO computes an update gradient at each step that minimizes the cost function while ensuring the deviation from the previous policy is relatively small. The new variant of the policy gradient is as follows:

$$J_{PPO}^{\theta_{old}} \approx \sum_{a_t, s_t} clip \left(\frac{p_\theta((a_t|s_t)}{p_{\theta_{old}}((a_t|s_t)}, 1-\varepsilon, 1+\varepsilon \right) A_{\theta_{old}}(a_t|s_t) \tag{6}$$

where θ_{old} is the parameter of the behavioral policy, p_θ is the distribution of the target policy, $p_{\theta_{old}}$ is the distribution of the behavioral policy, ε is the clipping coefficient. Then, the actor network can be updated by

$$\theta_{actor} \longleftarrow \theta_{actor} + \alpha \frac{\partial J_{PPO}^{\theta^k}}{\partial \theta_{actor}} \tag{7}$$

The critic network is updated by

$$\begin{aligned} \theta_{critic} \longleftarrow \theta_{critic} + \beta(r_{t+1} + \gamma V(s_{t+1}, \theta_{critic}) \\ - V(s_t, \theta_{critic})) \frac{\partial (V(s_t, \theta_{critic}))}{\partial \theta_{critic}} \end{aligned} \tag{8}$$

where α and β are the learning rates of actor network and critic network, respectively.

As shown in Fig. 1, the output state is also learned using the encoding network, but in the control stage, the actor-critic network (such as PPO) is utilized instead. In each step, the input image is encoded into a low-dimensional state vector by the trained model. Then, the critic network and actor network are updated by PPO to estimate the value function and learn the optimized control policy using the estimated policy gradient. For the continuous action space, we combine the encoding network and PPO to deal with the control problem in this paper and the details of our approach (SEN-DRL) are listed in Algorithm 2.

Algorithm 2 SEN-DRL for continuous actions

1: **Input:** a critic neural network and an actor neural network, a trained deep encoder network, a stop criterion.
2: **Output:** network parameters.
3: **Initialization:**
I: perceived image that capture the MDPs state;
β: the learning factor of the actor neural network;
t: the step in each learning episode and set $t = 0$.
4: **repeat**
5: Use the trained deep encoder network to encode the current perceived image I_t and utilize the encoded result s_t to calculate the output of the actor neural network y_t;
6: Output the actual action a_t according to the probability distribution;
7: Apply a_t to the MDPs and perceive the state transition (s_t, s_{t+1}) and compute reward $r_t = r\left(s_t, s_{t+1}\right)$;
8: Update the weights of the actor-critic neural network with Eq.(7) and Eq.(8)
9: Set $t = t + 1$
10: **until** the stop criterion is met.
11: RETURN network parameters.

Effectiveness analysis: In terms of the overall framework, the proposed SEN-DRL algorithm still conforms to the general paradigm of RL. The difference is that we introduce a pre-trained SEN at the input of the RL algorithm, so the performance of the policy learning is not

only determined by the subsequent RL algorithm but also influenced by the preceding state encoding network (SEN). Considering the prediction error of the encoding network, the SEN-DRL algorithm can be considered as RL under the condition that the state observations have noisy interference, *i.e.*, the RL algorithm receives the predicted state input as $o = s + \zeta$, where ζ can be regarded as the error of the state prediction. Thus, the learning performance of the algorithm will gradually become better as the noise intensity decreases as long as the encoding network can guarantee that the output state prediction error will decrease in the pre-training phase with the increase of training samples and the training times. The effectiveness of SEN-DRL algorithm can be guaranteed by integrating the state encoder network with state-based RL algorithms.

Sample efficiency analysis: Compared with the end-to-end DRL algorithms, the DRL algorithm based on pre-feature encoding splits the policy learning under the condition of high-dimensional state input into two stages: state feature encoder learning and RL based on encoded feature. On the one hand, through effective state encoding, the optimization space of the policy network will be substantially compressed. Taking the strategy optimization process as an example, if the state space $\mathcal{S}$ can be effectively reduced, the integration space required to solve the gradient can be significantly reduced, which will substantially reduce the interaction with the environment. On the other hand, the SEN is trained and designed with respect to the environment. Therefore, in the same environment, the pre-trained state coding network will not be affected even if the task objective changes and it can still complete the antecedent state encoding to assist the RL process.

4. Experimental Results

We evaluate our method on two benchmarks of learning control tasks, including the mountain-car task of discrete actions and the pendulum task of continuous actions. Additionally, the several typical RL/DRL methods are utilized for comparison. In the simulations, only the state-based RL methods such as SARSA, PPO (state) can obtain the real states as inputs while our SEN-DRL approach and other DRL algorithms such as DQN, DDQN, PPO (image), use the raw images as learning inputs which are high-dimensional. Both of the two tasks are based on the gym environment of OpenAI, and all of the implementation/training models are evaluated on pytorch and a Nvidia GeForce RTX 2080Ti GPU.

4.1 Mountain-Car Task

In this subsection, the image-based mountain-car task of discrete actions is used for performance tests. As shown in Fig. 3, the dynamical model of the mountain-car system follows the formula below [1]

$$\begin{aligned} p_{t+\Delta t} &= p_t + \dot{p}_t \\ \dot{p}_{t+\Delta t} &= \dot{p}_t + 0.001a - 0.0025cos(3p_t) \end{aligned} \tag{9}$$

where a is the discrete car's engine force and belongs to a discrete set [−1 N, 0 N, 1 N], and the position and velocity

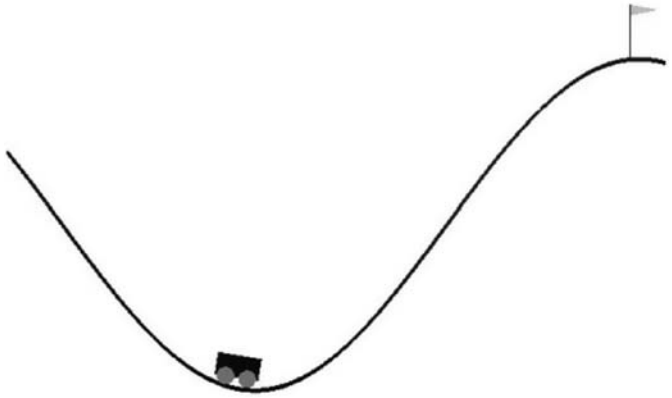


Figure 3. The mountain-car environment. The mountain-car task is to make the car, which is placed in the valley, drive to a preset destination (0.5 m in general) out of the valley in minimal steps. The car's engine is not powerful enough so that the car has to drive up opposing sides of the valley to accumulate enough momentum.

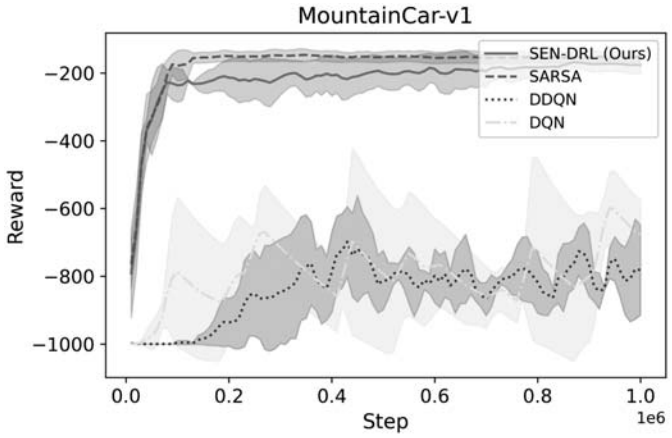


Figure 4. Comparison results of learning performance with the state-of-the-art approaches: SARSA, DQN and DDQN. The results of mean and variance in five runs are considered as the evaluation metrics. All of the approaches are trained with 1 million frames, and the max step is 1,000 during each episode.

are bounded in [−1.2 m, 0.5 m] and [−0.07 m/s, 0.07 m/s], respectively. When the car reaches the leftmost position, the velocity will be set to zero. In contrast, once it reaches the rightmost position, we state that the car has achieved the goal and the episode will be terminated. The state is a 2-D tuple $[p_t, \dot{p}_t]$.

The reward function is formulated as

$$R_t = \begin{cases} 0, & \text{if } s = s_G \\ -1, & \text{else} \end{cases} \tag{10}$$

where $s_G = 0.5\,\text{m}$ denotes the goal state.

During the experiment, we report the results after being trained with 1 million frames. Each episode starts at [−0.5 m, 0 m] and ends when the car reaches the goal state or after it takes 1,000 steps. For discrete-actions methods,

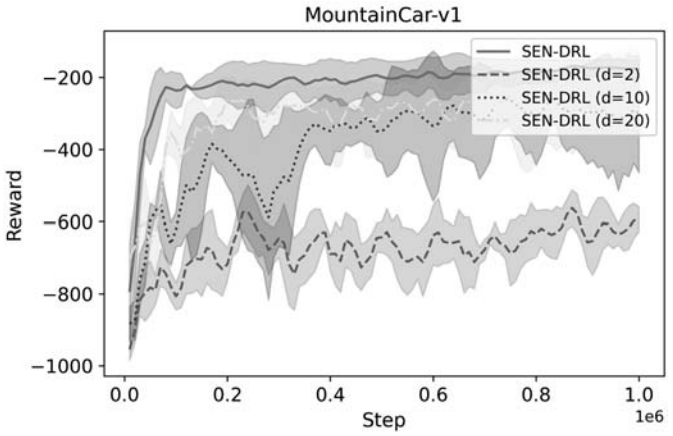


Figure 5. Comparison results of control performance using the policy learned in the mountain-car task under different numbers of training samples.

we choose three baselines for comparison: (1) SARSA. The standard SARSA considering the state as input. (2) DQN [6]: the standard DQN considering raw image as input. The Q network and Q-target network are represented by a fully connected MLP with three convolution layers and ReLU nonlinearities, respectively and (3) DDQN [34]. The standard DDQN considering raw image as input and the parameters are default. Figure 4 shows the learning performance among different approaches: SARSA, DQN and DDQN. Note that five independent runs are trained for each approach, all of the approaches are trained with 1 million frames, and the maximal step is 1,000 during each episode, and the curves in the figure stand for the average reward versus the step number. Note that the performance of SARSA is considered as the upper bound of other methods because its input is the accurate system state. It can be seen that it takes about 0.1 million to achieve a stable policy in our approach (SEN-DRL), indicating that our approach is competitive with SARSA. However, in both DQN and DDQN, the curves do not converge and the stable policies are not obtained. Although it might converge at some step (>1 million) in both DQN and DDQN, there is no need to verify it because our proposed approach has converged, which indicates that our approach performs the best compared with DQN and DDQN.

Intuitively, the number of training data during the state encoding process will affect the training results in SEN-DRL; to verify the generalization performance of our method, we evaluate the performance of SEN-DRL under different number of labeled samples shown in Fig. 5. The d denotes the encoding network is trained by d% proportion of the total samples. For example, SEN-DRL ($d = 20$) denotes the data are sampled as 20% proportion of the total samples in the mountain-car system. It can be seen that the SEN-DRL obtains better policy with the increase of the number of training data for CNN. Moreover, we can learn a good stable policy only using 10% of the total samples during encoder process, which indicates the generalization of our approach. To demonstrate the effectiveness of the proposed SEN, we evaluate the performance of SEN with different sparse settings over a successful trajectory with 183 timesteps in mountain-car task. The position and velocity prediction results are shown in Fig. 6. It can be seen that the approach with the different sample sizes can achieve similar accurate and stable position predictions in most states over the whole trajectory. However, for the velocity prediction, the sample number significantly affects, the prediction accuracy. Once we choose only 2% number of the samples, the result of the velocity is the worst. Therefore, to achieve better prediction results (especially in velocity prediction), it is important to choose an appropriate sample size, and we find 20% might be a good choice.

4.2 The Pendulum Task

In this subsection, the pendulum task of continuous actions is analyzed. As shown in Fig. 7, the pendulum starts at a random position and swings it up so it stays upright. The dynamic model of the pendulum system is defined as follows [44]:

$$\dot{\theta} = \dot{\theta} + \left(\frac{-3g}{2lsin(\theta\pi)} + \frac{3a}{ml^2}\right) dt \;\theta = \theta + \dot{\theta}dt \tag{11}$$

where $g = 9.8$ m/s, m is the mass of the pole and $m = 1$ kg, l is the length of the pole and $l = 0.5$ m, $dt = 0.5$. The state of the pendulum is $[\theta, \dot{\theta}]$, where θ is the angle between the pole and vertical direction, and $\dot{\theta}$ is the angular velocity. θ and $\dot{\theta}$ are bounded in $[-\pi,\pi]$ and $[-8$(rad/s), 8(rad/s)], respectively. For ease of algorithm design, the state is usually denoted as $[\cos\theta,\sin\theta,\dot{\theta}]$. The continuous action a of the pendulum is torque of joint, which is defined in $[-2$(N·m), 2(N·m)] and the reward function can be formulated as follows:

$$R = -(\theta^2 + 0.1\dot{\theta}^2 + 0.001a^2) \tag{12}$$

During our experiment, we report the image-based learning control results after being trained with 0.2 million frames. Each episode ends when the pendulum reaches the goal state or after it takes 200 steps, and the less force to keep the pendulum upright longer, the larger the reward. We compare our proposed SEN-DRL with two baselines: (1) PPO (state), the standard PPO with low-dimensional state as input, where the actor and critic are represented by a fully connected MLP with two hidden layers of 64 units and tanh nonlinearities. (2) PPO(image), the standard PPO with high-dimensional images as input. The actor and critic are represented by a fully connected MLP with three convolution layers and tanh nonlinearities. The parameters of PPO (state) and PPO (image) are similar as [2] and the implementation is based on stable-baselines [3]. Figure 8 shows the learning performance among different approaches: PPO (state) and PPO (image). Similar to the mountain-car task, five independent runs are tested for each method, all of the approaches are trained with 0.2 million frames, the maximal step is 200 during each episode, and the curves in the figure stand for the average reward versus the step number. Here, PPO (state) is considered as the upper bound of other methods because its input is the accurate state of the system. It can be seen that it takes about 0.075 million to achieve a stable

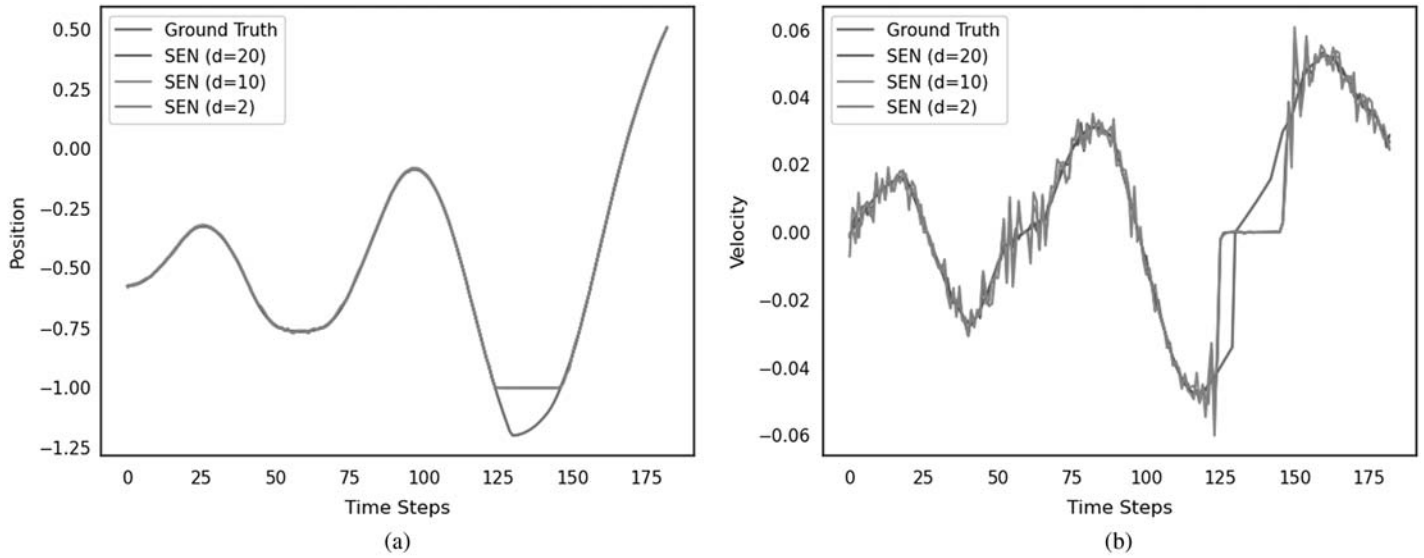


Figure 6. The comparison of state encoding under different number of labeled samples in the mountain-car task.

Figure 7. Pendulum system, which consists of a pole and the goal of the task is to swing the pole up so it stays upright.

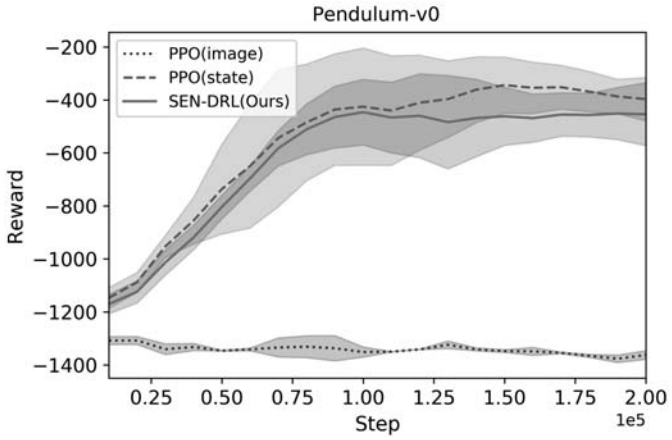


Figure 8. Comparison results of learning performance with the state-of-the-art approaches: PPO (state) and PPO (image). The results of mean and variance in 5 runs are considered as the evaluation metrics. All of the approaches are trained with 0.2 million frames, and the max step is 200 during each episode. In both PPO (state) and our methods, it takes about 0.075 million steps to achieve a stable policy, indicating that our method is competitive with PPO (state). However the curve does not converge and the stable policy is not obtained in PPO (image), which indicates that our approach outperforms the PPO (image).

policy in our approach (SEN-DRL), indicating that our approach is competitive with PPO (state). However, in PPO (image), the curve does not converge and no stable policy is obtained. Although it might converge at some step (>0.2 million), there is no need to verify it because our proposed approach has converged, which indicates that our approach performs better than PPO (image).

To verify the generalization of the number of labeled data of SEN-DRL, we record the learning performance of SEN-DRL under different number of labeled samples shown in Fig. 9. The d corresponds to the CNN encoder trained by d% samples of the standard SEN-DRL. It can be seen that the SEN-DRL adapts to the number of training data to some extent, but when the training data for CNN encoder are too sparse, SEN-DRL might not learn a stable policy.

To demonstrate the effectiveness of the proposed SEN, we evaluate the performance of SEN with different sparse settings over a trained policy in pendulum task. The theta and angular velocity prediction results are shown in Fig. 10. It can be seen that the approach with the different sample sizes can achieve similarly accurate and stable theta predictions in most states over the whole trajectory. However, in the angular velocity prediction case, the sample number significantly affects the prediction accuracy. Once we choose only 2% number of the samples, the result is the worst. Therefore, to achieve better prediction results (especially in angular velocity prediction), it is important to choose an appropriate sample size, and we find 20% might be a good choice.

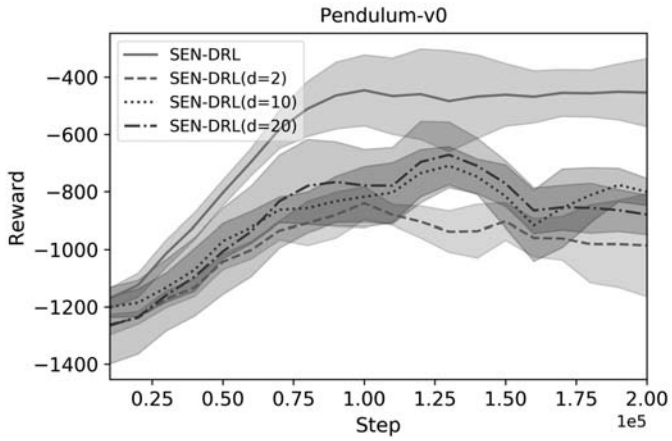


Figure 9. Comparison results of control performance using the policy learned in the pendulum task under different numbers of training samples.

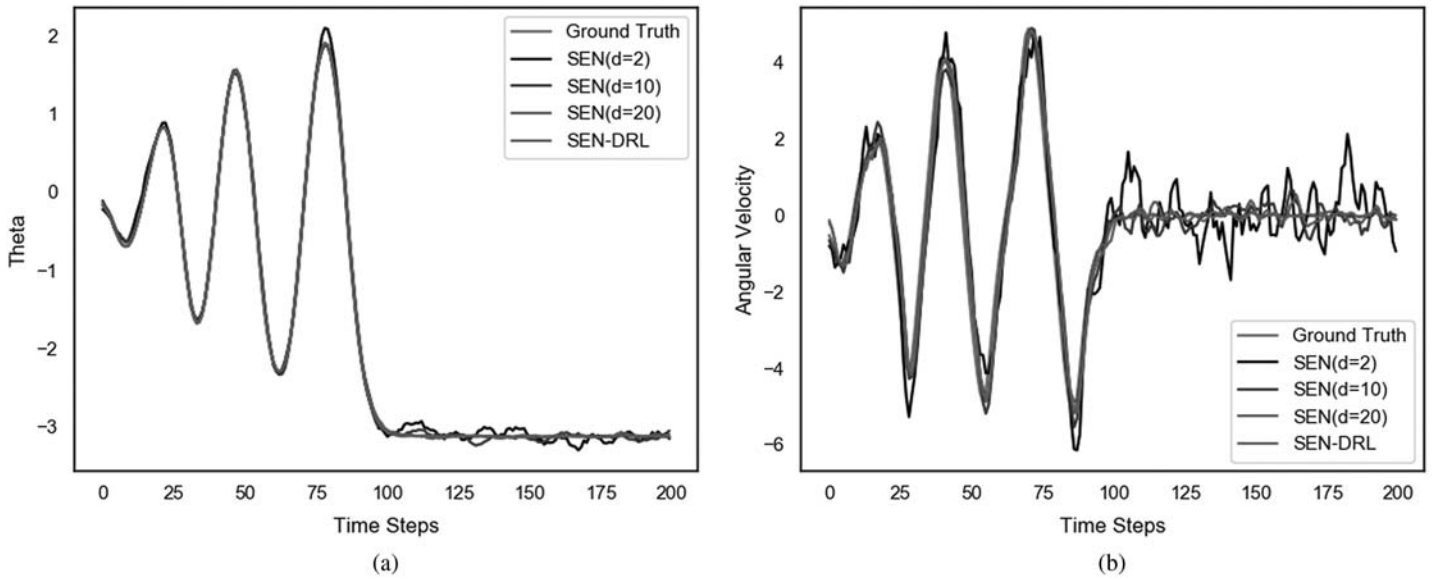


Figure 10. The comparison of state encoding under different number of labeled samples in the pendulum task.

5. Conclusions

To improve data efficiency of DRL algorithms, this paper presents a cascaded RL approach which can realize fast online end-to-end learning by mapping the raw image data into lower-dimensional states to improve learning efficiency and convergence speed. A convolution encoding network is used to encode features from raw image data, so that online DRL algorithms can be performed based on the encoded low-dimensional states. Simulation results on two benchmark of learning control tasks demonstrate that the proposed approach is more general and efficient compared with previous DRL methods, such as DQN, DDQN and PPO. Further developments of the proposed cascaded RL approach and applying it to real-world learning control problems are our ongoing work.

Funding

This work was supported by the National Natural Science Foundation of China (grant nos. 61825305 and 61703418).

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, 2018).

[2] X. Yang, H.B. He, and Q.L. Wei, Reinforcement learning for robust adaptive control of partially unknown nonlinear systems subject to unmatched uncertainties, *Information Sciences*, *463*, 2018, 307–322.

[3] S.S. Chong, L.P. Wong, and C.P. Lim, Automatic design of hyper-heuristic based on reinforcement learning, *Information Sciences*, *245*, 2018, 89–107.

[4] X. Xu, Z.H. Huang, L. Zuo, and H.B. He, Manifold-based reinforcement learning via locally linear reconstruction, *IEEE Transactions on Neural Networks and Learning Systems*, *28*(4), 2017, 934–947.

[5] D.B. Zhao, D.R. Liu, and F.L. Lewis, Special issue on deep reinforcement learning and adaptive dynamic programming, *IEEE Transactions on Neural Networks and Learning Systems*, *29*(6), 2018, 2038–2041.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, *et al.,* Human-level control through deep reinforcement learning, *Nature*, *518*(7540), 2015, 529–C533.

[7] G.B. Huang, Q.Y. Zhu, and C.K. Siew, Extreme learning machine: Theory and applications, *Neurocomputing*, *70*(1–3), 2006, 489–501.

[8] Y.F. Wei, F.R. Yu, and M. Song, User scheduling and resource allocation in Hetnets with hybrid energy supply: An actor-critic reinforcement learning approach, *IEEE Transactions on Wireless Communications*, *17*(1), 2018, 680–692.

[9] X. Xu, H.G. He, and D.W. Hu, Efficient reinforcement learning using recursive least-squares methods, *Journal of Artificial Intelligence Research*, *16*, 2002, 259–292.

[10] H.Z. Wang, Y.L. Wu, and G.Y. Min, Data-driven dynamic resource scheduling for network slicing: A deep reinforcement learning approach, *Information Sciences*, *498*, 2019, 106–116.

[11] J. Baxter, P.L. Bartlett, and L. Weaver, Experiments with infinite-horizon, policy-gradient estimation, *Journal of Artificial Intelligence Research*, *15*(1), 2001, 351–381.

[12] J. Ni, X. Li, M. Hua, and S.X. Yang, Bio inspired neural network based q-learning approach for robot path planning in unknown environments, *International Journal of Robotics and Automation*, *31*(6), 2016, 464–474.

[13] T. Yan, W. Zhang, S.X. Yang, and L. Yu, Soft actor-critic reinforcement learning for robotic manipulator with hindsight experience replay, *International Journal of Robotics and Automation*, *34*(5), 2019, 206–216.

[14] J.H. Liu, X. Xu, and Z.H. Huang, Model-free multi-kernel learning control for nonlinear discrete-time systems, *International Journal of Robotics and Automation*, *32*(5), 2017, 401–410.

[15] Z. Chu, D. Zhu, and S.X. Yang, Observer-based adaptive neural network trajectory tracking control for remotely operated vehicle, *IEEE Transactions on Neural Networks and Learning Systems*, *28*(7), 2016, 1633–1645.

[16] Y. Liu, M. Cong, and H. Dong, Reinforcement learning and ega-based trajectory planning for dual robots, *International Journal of Robotics and Automation*, *33*(4), 2018, 140–149.

[17] N.T. Luy, T. Nguyen, and H.M. Tri, Reinforcement learning-based intelligent tracking control for wheeled mobile robot, *IEEE Transactions on the Institute of Measurement and Control*, *36*(7), 2014, 868–877.

[18] G.X. Feng, L. Busoniu, T.M. Guerra, and S. Mohammad, Data-efficient reinforcement learning for energy optimization of power-assisted wheelchairs, *IEEE Transactions on Industrial Electronics*, *66*(12), 2019, 9734–9744.

[19] C. Wu, Y.M. Wang, and Z.J. Yin, Realizing railway cognitive radio: A reinforcement base-station multi-agent model, *IEEE Transactions on Intelligent Transportation Systems*, *20*(4), 2019, 1452–1467.

[20] S. Tetsuya, S. Kyoko, M. Tadanobu, and O. Yoshitaka, Predicting investment behavior: An augmented reinforcement learning model, *Neurocomputing*, *72*(16–18), 2009, 3447–3461.

[21] W.S. Dean and K.D. George, Regularized feature selection in reinforcement learning, *Machine Learning*, *100*(2–3), 2015, 655–676.

[22] Y. Bengio, A. Courville, and P. Vincent, Representation learning: A review and new perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 2013, 1798–1828.

[23] X. Xu, D.W. Hu, and X.C. Lu, Kernel-based least squares policy iteration for reinforcement learning, *IEEE Transactions on Neural Networks*, *18*(4), 2007, 973–992.

[24] L. Yang, Y. Lu, S.X. Yang, T. Guo, and Z. Liang, A secure clustering protocol with fuzzy trust evaluation and outlier detection for industrial wireless sensor networks, *IEEE Transactions on Industrial Informatics*, *7*(7), 2021, 4837–4847.

[25] Z. Ren, S.X. Yang, Q. Sun, and T. Wang, Consensus affinity graph learning for multiple kernel clustering, *IEEE Transactions on Cybernetics*, *51*(6), 2021, 3273–3284.

[26] H.L. Li, D.R. Liu, and D. Wang, Manifold regularized reinforcement learning, *IEEE Transactions on Neural Networks and Learning Systems*, *29*(4), 2018, 932–943.

[27] S.P. Zhao, B. Zhang, and C.L. Philip, Joint deep convolutional feature representation for hyperspectral palmprint recognition, *Information Sciences*, *489*, 2019, 167–181.

[28] C. Hodges, M. Bennamoun, and H. Rahmani, Single image dehazing using deep neural networks, *Pattern Recognition Letterss*, *128*(9), 2019, 70–77.

[29] J. Gan, W.Q. Wang, and K. Lu, A new perspective: Recognizing online handwritten Chinese characters via 1-dimensional CNN, *Information Sciences*, *478*, 2019, 375–390.

[30] Y.Y. Chen, J.Q. Wang, B.K. Zhu, M. Tang, and H.Q. Lu, Pixelwise deep sequence learning for moving object detection, *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(9), 2019, 2568–2579.

[31] B.S. Dunja, Z. Marusic, and S. Gotovac, Deep learning approach in aerial imagery for supporting land search and rescue missions, *International Journal of Computer Vision*, *127*(9), 2019, 1256–1278.

[32] X.D. Li, M. Ye, Y.G. Liu, and C. Zhu, Adaptive deep convolutional neural networks for scene-specific object detection, *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(9), 2019, 2538–2551.

[33] U. Raghavendra, H. Fujita, S.V. Bhandary, A. Gudigar, and U.R Acharya, Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images, *Information Sciences*, *441*, 2018, 41–49.

[34] H.V. Van, A. Guez, and D. Silver, Deep reinforcement learning with double q-learning, in *AAAI*, vol. 2 (2016), 2094–2100.

[35] G. Lever N. Heess, D. Silver, and M. Riedmiller, *Deterministic policy gradient algorithms* (2014).

[36] S. Levine, P. Abbeel, M. I. Jordan, J. Schulman, and P. Moritz, *Trust region policy optimization* (2015), 1889–1897.

[37] F. Wolski, P. Dhariwal, A. Radford, J. Schulman, and O. Klimov, Proximal policy optimization algorithms. *arXiv preprint*, page arXiv (2017).

[38] T. Hester and P. Stone, Intrinsically motivated model learning for developing curious robots, *Artificial Intelligence*, *247*, 2017, 170–186.

[39] J. Oh, S. Singh, and H. Lee, Value prediction network, in *Advances in Neural Information Processing Systems* (2017), 6118–6128.

[40] M. Babaeizadeh, L. Frosio, S. Tyree, J. Clemons, and J. Kautz. Reinforcement learning through asynchronous advantage actor-critic on a GPU. *arXiv* (2016).

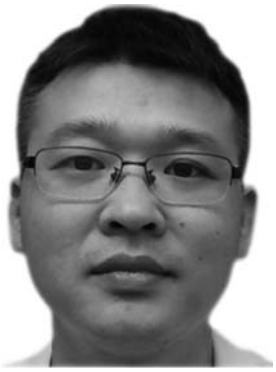
[41] L. Tai and M. Liu, Mobile robots exploration through CNN-based reinforcement learning, *Robotics and Biomimetics*, *3*(1), 2016, 1–8.

[42] L. Szoke, S. Aradi, T. Becsi, and P. Gaspar, Driving on highway by using reinforcement learning with CNN and LSTM networks, in *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)* (2020).

[43] H. Bae, G. Kim, and J. Kim, Multi-robot path planning method using reinforcement learning, *Applied Sciences*, *9*(15), 2019, 2076–2090.

[44] K.J. Astrom and K. Furuta, Swinging up a pendulum by energy control, *Automatica*, *36*(2), 2000, 287–295.

Biographies



Qiang Fang received the B.S. degree in automation from the Xidian University, Xi'an, China, in 2007. He received the M.S. and Ph.D. degrees in control science and engineering from the National University of Defense Technology, Changsha, China, in 2009 and 2013, respectively.

He is currently an Associate Professor with the College of Intelligence Science and Engineering, National University of Defense Technology. His research interests include robotics and unmanned aerial vehicles, with focus on autonomous navigation, deep learning and reinforcement learning.

Xin Xu (M'09-SM'12) received the B.S. degree in electrical engineering from the Department of Automatic Control, National University of Defense Technology (NUDT), Changsha, China, in 1996, and the Ph.D. degree in control science and engineering from the College of Mechatronics and Automation, NUDT, in 2002.

He has been a Visiting Professor with Hong Kong Polytechnic University, Hong Kong; the University of Alberta, Edmonton, AB, Canada; the University of Guelph, Guelph, ON, Canada; and the University of Strathclyde, Glasgow, U.K. He is currently a Professor with the College of Intelligence Science and Technology, NUDT. He has coauthored over 180 papers in international journals and conferences, and coauthored four books. His current research interests include intelligent control, reinforcement learning, approximate dynamic programming, machine learning, robotics and autonomous vehicles. Dr. Xu was a recipient of the National Science Fund for Outstanding Youth in China and the Second-Class National Natural Science Award of China. He has served as an Associate Editor for Information Sciences, IEEE Transactions on Systems, Man, and Cybernetics: Systems, International Journal of Robotics and Automation, Intelligent Automation and Soft Computing, and Acta Automatica Sinica. He is a member of the IEEE CIS Technical Committee on Approximate Dynamic Programming and Reinforcement Learning and the IEEE RAS Technical Committee on Robot Learning.

Yixing Lan received the M.S. degree from National University of Defense Technology, Changsha, China, in 2018. He is currently pursuing the Ph.D. degree. His current research interests include reinforcement learning, transfer learning and multi-agent reinforcement learning.

Yichuan Zhang a Master student at the College of Intelligence Science and Technology, National University of Defense Technology. He received his bachelor degree from National University of Defense Technology in 2019. His research interest covers reinforcement learning, imitation learning and bayesian models.

Yujun Zeng was born in Jiangxi, China, in 1990. He received the Automation degree, the M.S. degree and the Ph.D. degree in Pattern Recognition and Intelligent System in National University of Defense Technology, Changsha, Hunan, China, in 2011, 2014 and 2018 respectively. Now he is a Research Assistant with the Pattern Recognition and Intelligent System Laboratory of the College of Intelligence Science and Technology, National University of Defense Technology. His research interests include computer vision, pattern recognition, neural networks and machine learning.

Tao Tang received his Bachelor Degree from National University of Defense Technology, Changsha, China, in 2020. He is currently pursuing the master degree in the same university in the direction of machine learning. His current research interests include reinforcement learning, transfer learning and meta-learning.