Robot Navigation in Crowds Environment Base Deep Reinforcement Learning with POMDP

Qinghua Li[1,3], Haiming Li[1,3], Jiahui Wang[1,3], and Chao Feng[2,3(✉)]

[1] School of Information and Automation, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, People's Republic of China
[2] International School for Optoelectronic Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, People's Republic of China
cfeng@qlu.edu.cn
[3] Jinan Engineering Laboratory of Human-Machine Intelligent Cooperation, Jinan 250353, People's Republic of China

Abstract. With the development of deep learning technology, the navigation technology of mobile robot based on deep reinforcement learning is developing rapidly. But, navigation policy based on deep reinforcement learning still needs to be improved in crowds environment. The motion intention of pedestrians in crowds environment is variable, and the current motion intention information of pedestrian cannot be judged by only relying on a single frame of sensor sensing information. Therefore, in the case of only one frame of input, the pedestrian motion state information is partially observable. To dealing with this problem, we present the P-RL algorithm in this paper. The algorithm replaces traditional deep reinforcement learning Markov Decision Process model with a Partially Observable Markov Decision Process model, and introduces the LSTM neural network into the deep reinforcement learning algorithm. The LSTM neural network has the ability to process time series information, so that makes the algorithm has the ability to perceive the relationship between the observation data of each frame, which enhances the robustness of the algorithm. Experimental results show our algorithm is superior to other algorithms in time overhead and navigation success rate in crowds environment.

Keywords: Deep reinforcement learning · Robot navigation · Partially observable Markov decision process

1 Introduction

With the development of technology that automatic driving and artificial intelligence, the application scene of the robot has been expanded from the industrial environment to the social environment of sharing activity space with human. Mobile robot navigation is used in factories, hospitals and shopping malls. These

S.-H. Wang and Y.-D. Zhang (Eds.): ICMTEL 2022, LNICST 446, pp. 675–685, 2022.
https://doi.org/10.1007/978-3-031-18123-8_53

tasks are still a challenging problem, because they require mobile robots to navigate safely and effectively in crowds environment [7,9,14].

Collision avoidance is an essential ability which the mobile robots navigate in crowds environment. Early works have proposed many methods which based on pedestrian motion model to dealing with the problem of mobile robot navigation in crowds environment. The pedestrian motion models have three main categories: social force model, data-driven and geometric approaches. The social force model [6,8] proposes a model of crowds interaction based on Gaussian process. These methods perform well in a crowds simulation, but they usually do not predict the movement of individual pedestrian. The data-driven approaches [1,13] can learn pedestrian dynamics from past trajectories. But these methods usually hard to obtain the required training data and the learning model may be not well generalized to different scenarios. The geometric approaches include the Reciprocal Velocity Obstacle(RVO) [2] and the Optimal Reciprocal Collision Avoidance(ORCA) [11]. This kind of methods through optimization the geometric feasible space to calculates the obstacle avoidance paths for multi-agents. But, these methods can not understand the diversity of human behavior, the movement trajectory of robot is short-sighted in time which lead to unnatural robot behaviors and create movement oscillatory in crowds environment [3].

With the development of technology which artificial intelligence and automatic driving. Robots encode features related to the interaction between the crowds or robots in the navigation policy and use neural network learning experience to understand crowds environment which produces paths that are close to human behavior through learning. Many researches have proposed the methods which motion planner base on deep reinforcement learning [10,15]. These methods learn policies from raw sensor input of the environment by reinforcement learning methods. However, it is difficult to extract the richer high-level representation of pedestrian intention in raw sensor information which makes them difficult navigation results in crowds environments. In order to dealing with this problem, some people proposed the deep reinforcement learning navigation algorithm based on the representation of pedestrian state which integrates pedestrian motion prediction into the decision making process to generate a path close to human behavior [3–5]. The algorithm extracts pedestrian state information from the original data as the input of reinforcement learning. The relevant features of the interaction between the crowds and robot are encoded into a fixed-length vector, which processes the state of each pedestrian in descending order according to the distance which between the robot and pedestrians. Although these methods have proved well results when working in a crowds environment, but there are still some limitations for robot navigation. These methods are based on single frame data and do not process the time series information, which will lead to short-sighted and make a detour of robots in crowds, because of they do not consider the future movement state of pedestrians.

In this work, we propose a new algorithm which can dealing with these previous shortcomings. Inspired by DRQN algorithm [12], our algorithm will consider: First, we incorporated the interactions between the observation data of adjacent

frames into the reinforcement learning network to overcome the short-sighted problem of robot trajectory time, and without using multi frame data class to predict the future pedestrian trajectory. Second, we use of the POMDP model to replace with the MDP model in deep reinforcement learning, which enhance performance whith robustness of the algorithm. Thirdly, we add the attention network into the neural network, infer the relative importance of the adjacent frame data relative to its future state through the attention network, so as to focus on the key frame data and improve the learning efficiency.

This paper is structured as following. In the second section, we introduces the related work of robot navigation algorithm. In the third section, we introduces the robot navigation base on deep reinforcement learning. Introduces the details of the P-RL algorithm in the fourth section. Introduces the experiments and result in the fifth section. Finally we concluded the algorithm.

2 Related Work

Early works have proposed many methods which based on pedestrian motion model to dealing with the problem of robot navigation in crowds environment. The Optimal Reciprocal Collision Avoidance(ORCA) [11] is the best performance algorithm in these algorithms. In this algorithm, the robot calculates the velocity space of other agents to avoid collisions with them, the robot can selects the optimal velocity in the intersection of all permitted geometric feasible velocity space. However, since the velocity space-based method does not consider the change of future state with agents, they will create oscillatory and unnatural behaviors in crowds environment.

The method of using pedestrian motion model to dealing with the problem of mobile robot in dense crowded is too depend on the human-engineered hyper parameters and rules which the effect is often poor. In order to dealing with these problems, deep reinforcement learning method has been widely studied in the field of robot navigation. In Deep reinforcement learning method, the robot use neural network learning experience to understand crowds environment which produces paths that are close to human behavior through learning.

There are a number of recent studies proposed the collision avoidance algorithm using deep reinforcement learning which integrates pedestrian motion prediction into the decision-making process to generate a path close to human behavior [3–5]. Chen proposed a collision avoidance algorithm [4,5], this methods extracts the pedestrian motion state information from the original data and takes the pedestrian motion state information as the input of reinforcement learning. The relevant features of the interaction between the crowds and robot are encoded into a fixed-length vector, which processes the state of each pedestrian in descending order according to the distance between the pedestrian and the robot. However, it is not reasonable to allocate importance according to distance, the pedestrian following the robot may not be as important as the farther pedestrian in front of it. A recent work proposed a approaches named SARL [3]. This method improved on previous work which uses the self-attention module

to allocate different importance weights to different parts of the crowds, so as to further improve the navigation performance.

Although these methods have proved well results when working, but there are still some shortcomings for robot navigation. Firstly, these methods are based on an idealized assumption that the motion states of robots and pedestrians are globally known and absolutely real. And, these methods are based on single frame data and do not process the time series information. It will lead to the robots short-sighted and make a detour in the dense crowds because of they do not consider the future movement state of pedestrians.

3 Robot Navigation Base Deep Reinforcement Learning

3.1 Problem Formulation

We designed a task of robot navigation that robot through the crowds and move to a random goal position. The navigation task of mobile robot in the crowds can be regarded as a decision process problem in deep reinforcement learning. We take crowds environment as the multiple agent problem to modeling between the robot and the crowds, where only robot agents' policy π is trainable, and the crowds agents' policy $\tilde{\pi}$ is designed to a unknown function that is modeled as a part of the environment.

In the past research, it is usually assumed that there is the robot and crowds in the X-Y plane of the 2D-workspace. They suppose each agent state can be observed, it include the position p, velocity v, orientation θ, goal position g, preferred speed v_{pref} and radius r, which are expressed as $p = [p_x, p_y]$, $v = [v_x, v_y]$, θ, r, $g = [g_x, g_y]$ and v_{pref}. The robot state at time t can be defined as $S_t = [p_x, p_y, v_x, v_y, \theta, g_x, g_y, r, v_{pref}]$, and the crowds state at time t of each human state can be defined as $O_t^i = [p_x^i, p_y^i, v_x^i, v_y^i, \theta, g_x^i, g_y^i, r^i, v_{pref}^i]$. In the real-world, it is difficult for sensors to perceive the absolute real motion state of pedestrian based on single frame data. So as to simulate the real-world environment, we remove the speed, goal position and preferred speed information of each crowds agent, and assume that the robot only observe the position, direction and radius of each crowds agent. After modify, the robot data of each frame become that states $S_t = [p_x, p_y, v_x, v_y, \theta, g_x, g_y, r, v_{pref}]$ and observation state $O_t^i = [p_x^i, p_y^i, \theta, r^i]$.

At each time step t, the robot observes a state of crowds which is include the robot state and the each humans motion states. It can be defined as $S_t^{jn} = [S_t, O_t^1, O_t^2, ..., O_t^n]$.

The reward $R(s_t^{jn}, a_t)$ is designed to excitation the robot when reach goal position and punish the robot close to humans or collisions humans.

$$R(s_t^{jn}, a_t) = \begin{cases} -0.25 & d_{min} < 0, \\ -0.1 + d_t/2 & 0 < d_{min} < 0.2, \\ 1 & p_t = p_g, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where d_{min} is the minimum safe distance between humans and robot for a cycle, $p_t = [p_x, p_y]$ is the robot position at each time step t, $p_g = [g_x, g_y]$ is the goal position.

We expect to obtain an optimal policy π^* by deep reinforcement learning that maximizes the expectation of discounted total rewards, the optimal value function V^* of the state S_t^{jn} can be formulated as:

$$V^*(S_t^{jn}) = \sum_{t=0}^{T} \gamma^t \cdot R\left(s_t^{jn}, \pi^*\left(s_t^{jn}\right)\right), \tag{2}$$

where $\gamma \in [0, 1)$ is a discount factor.

Base on Bellman Equation, the optimal value function can get a optimal policy $\pi^*\left(s_t^{jn}\right)$ by the value iteration method. It can be derived as:

$$\begin{aligned} \pi^*(S_t^{jn}) = & \underset{a_t \in A}{argmax} R(s_t^{jn}, a_t) + \\ & \gamma^{\triangle t} \int\limits_{S_{t+\triangle t}^{jn}} P(S_{t+\triangle t}^{jn} | S_t^{jn}, a_t) V^*(S_{t+\triangle t}^{jn})\, dS_{t+\triangle t}^{jn} \end{aligned} \tag{3}$$

where $\triangle t$ is the time of decision interval between two actions, A is the action space, $P(S_{t+\triangle t}^{jn} | S_t^{jn}, a_t)$ is a transition probability from S_t^{jn} to $S_{t+\triangle t}^{jn}$ when the action a_t is executed.

3.2 Partially Observable Markov Decision Processes

In the real environment, Markov property is difficult to hold. The Partially Observable Markov Decision Process model is an extension of Markov Decision Process in partially observable environment, it can better capture the dynamics of many real-world environments. Typically, the POMDP model can be defined as the tuple (S, A, T, R, Z, O).

S is a collection of real state s_t in the real environment.

A is a collection of all available actions of the robot, and $a_t \in A$ stand for the action which the robot take in time t.

T is the probability distribution of the agent transferring to other states s_{t+1} after executing action a_t at the state s_t, $T(s_t, a_t, s_{t+1}) = p(s_{t+1}|s_t, a_t)$.

R is the reward which represents the reward after the agent takes an action a_t, and it can be defined as $R(s_t, a_t) = r_t$.

Z is a collection of observation results, which is the environmental data obtained by the robot's sensors.

O is the probability distribution of receiving observation z_t after the agent take action a_t in the state s_t, $O(s_t, a_t, z_t) = p(z_t|s_t, a_t)$.

In the partially observable markov decision processes system, the system state is not completely known. We maintain a belief over possible states. It defined a

belief update function τ to estimate the current state.

$$b_t(s_{t+1}) = \eta O(s_t, a_t, z_t) \sum_{s_t \in S} T(s_t, a_t, s_{t+1}) b_{t-1}(s_t) \tag{4}$$

$$b_t = \tau(b_0, a_1, z_1, a_2, z_2, ..., a_t, z_t) \tag{5}$$

where η is a normalizing constant, b_0 is an initial belief, $s_t \in S$, $b(s_t) \geqslant 0$, and $\sum_{s_t \in S} b(s_t) = 1$.

4 P-RL

4.1 Overall Framework

In this paper, we inspired by the DRQN algorithm [12] and proposed a new algorithm. In our algorithm, we use of POMDP model to replace the MDP model in reinforcement learning, and introduces the LSTM neural network into the Value-Network. It deal with the problem of the robot navigation unsafe and detour in the crowds environments. This algorithm as outlined in Algorithm 1.

Algorithm 1: P-RL algorithms

```
Initialize Value-Network Q;
Initialize the observation space S, action space A;
Set a random goal position g;
while Goal not reached do do
    Get a observation state of robot and crowds S_t^{jn};
    Select a_t = argmax_{a_t ∈ A} R(s_t^{jn}, a_t) + γ^{△t·v_pref} · Q(s_t^{jn}), Execute action a_t;
    Obtain reward r_t and result observation s_{t+1}^{jn};
end
```

4.2 Value-Network

In order to make the algorithm have the ability to process time series, we introduce the LSTM neural network and attention module into the Value-Network. We divide the Value-Network into feature module, belief module and decision module, as show in Fig. 1.

Feature Module. We used the multi-layer perceptron neural networks(MLP1) to encode state of robot and crowd into a fixed length vector, and obtain the high-dimensional feature information of human-robot interaction ε_t.

$$\varepsilon_t = \varphi_\varepsilon(s_t^{jn}; W_\varepsilon) \tag{6}$$

where $\varphi_\varepsilon(\cdot)$ is an encoded function composed of a multi-layer perceptron with relu activations function, W_ε is the encode weights.

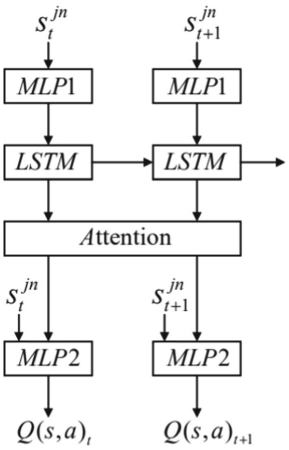

Fig. 1. The Value-Network is made up of the multi-layer perceptron neural networks with activations function Relu and long-short memory networks and attention module. It can be divided into feature module, belief module and decision module. The inputs is the robot and crowds motion state, and the outputs is a optimal value of the value function.

Belief Module. The module is made up of the LSTM neural network and the attention module. In this module, the LSTM neural network infers the relationship between each frame's feature information of human-robot interaction h_t. The attention module scored each frame, and weighted by the attention score α_t to find the keyframe. Since the module has strong inference ability to time series, we use it as the belief update function of POMDP model. The module input is the feature information of human-robot interaction for each frame, and output the corresponding belief b_t.

$$h_t = \psi_h(\varepsilon_t) \tag{7}$$

where $\psi_h(\cdot)$ is a time series inference function composed of a long-short memory network.

$$\alpha_t = \phi_\alpha(h_t; W_\alpha) \tag{8}$$

where $\phi_\alpha(\cdot)$ is an attention function composed of a multi-layer perceptron.

$$b_t = \sum_{t=0}^{T} softmax(\alpha_t) h_t \tag{9}$$

Decision Module. The module a multi-layer perceptron(MLP2). This module is a regression network, which is composed of multiple fully connected layers. It can be predicted the Q-value through observation state S_t^{jn} and belief value b_t.

$$Q_t = \mathcal{V}_Q(S_t^{jn}, b_t; W_Q) \tag{10}$$

where $\mathcal{V}_Q(\cdot)$ is a decision function composed of a multi-layer perceptron, W_Q is the decision weights.

5 Experiments

5.1 Simulation Setup

We built a simulation environment based on Gym which can simulation robot navigation in crowds environment. In this simulation environment, we used circles with the radius of 0.25 m to behalf of robots and crowds. Among them, robots for crowds are controlled by Optimal Reciprocal Collision Avoidance algorithm [11]. We assume that the robot and the crowd move freely in a space, in which the crowd will not actively avoid the robot. The robot and the crowd are randomly placed on a circular side length with a diameter of 4 m, and the opposite point on the side length over the center of the circle is set as the goal position. The goal of robots and crowd was to move from the start position to the goal position. So as to improve the training efficiency, we set the upper limit of robot navigation time to 25 s. It is defined as navigation failure, when the navigation time exceeds this upper limit.

5.2 Quantitative Evaluation

There are three most advanced methods, CADRL [5], LSTM-RL [4] and SARL [3], are implemented as base-line methods for this experiments. It is difficult to obtain the information of pedestrian speed in the simulated real world, so we delete the information of speed in the population state based on the baseline method. So as to assure the equity of the experiment, we run all algorithms in this environment.

We use the method with Temporal-Difference Learning to train the deep reinforcement learning algorithm. We create some data sets by ORCA [11] algorithm to pretreatment train the neural network before deep reinforcement learning training. We generate 2000 pretrain data with ORCA [11] to initialization of deep reinforcement learning. After pretraining, we train the algorithm for 4000 times of reinforcement learning. We use the method of dynamic greedy coefficient ε to improve the learning efficiency of deep reinforcement learning. At the initial stage of training, the greedy coefficient e is set to 0.5, which makes the algorithm more exploratory to generate more training data. With the increase of training times, we gradually reduce the greedy coefficient ε to 0.01, making the algorithm trust the trained neural network to promote the convergence of the algorithm.

Figure 2 shows the curves of success rate, collision rate, reward and cost time to reach goal in the simulated environment after 4000 rounds of training for CADRL [5], LSTM-RL [4], SARL [3] and P-RL.

Analyzing Fig. 2, we can conclude that in the training environment without pedestrian speed information, all base-line methods have been affected to a certain extent. Among them, CADRL [5] can not converge without pedestrian speed

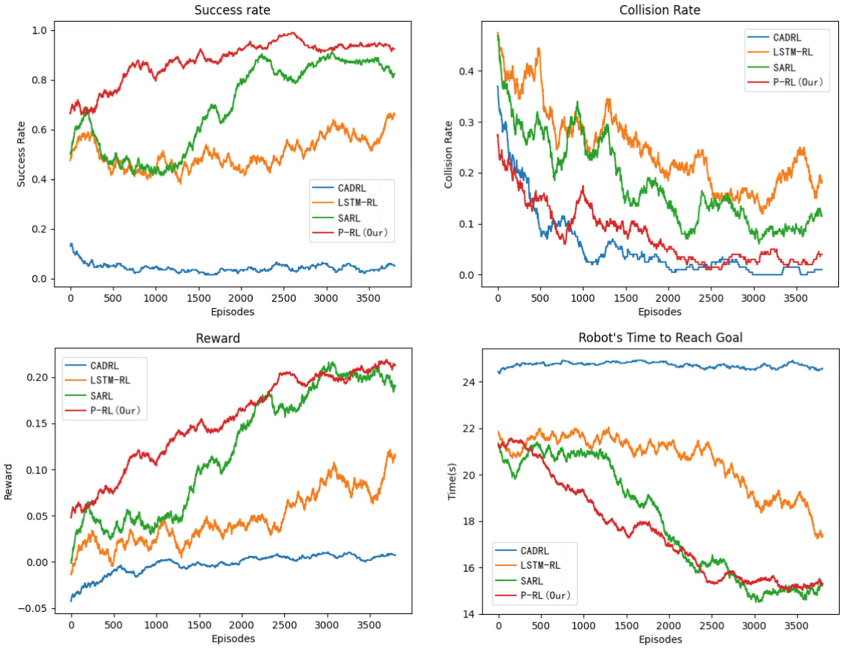

Fig. 2. The curves of success rate, collision rate, reward and cost time to reach goal in the simulated environment after 4000 rounds.

information. In collision rate, LSTM-RL [4] and SARL [3] can not achieve good convergence, and the collision rate remains at a high level. Although CADRL [5] has the lowest collision rate, but the success rate and cost time is basically zero. It can be seen that CADRL [5] maintains excellent collision avoidance ability, but it has lost its path planning ability, so it is unable to complete the robot mobile task. Compared with the base-line methods, our proposed algorithm P-RL can maintain better navigation performance even when the pedestrian speed information is lost. From the graph, we can see that our algorithm maintains high robustness in terms of Success rate, Collision rate and Navigation Time. And through the reward curve, we can see that our algorithm maintains a higher learning efficiency than base-line methods, it can learn better navigation strategies faster than base-line methods.

We used 500 random navigation tests to evaluate the model, using the average of 500 tests data as the evaluation benchmark, as show in Table 1.

Analyzing Table 1, we can conclude that in the partially known environment where the pedestrian speed state is lost, the data of the baseline algorithm can not meet the standard, and the success rate is lower than 0.8, so the robot navigation task can not be completed well. We provide the P-RL algorithm has strong robustness, it still can maintain a success rate of 0.9 in this environment, and the collision rate and navigation time are better than the baseline algorithm. This shows that our algorithm has better performance in robot navigation tasks.

Table 1. Table of the average of 500 random navigation tests data.

Method	Success rate	Collision rate	Navigation time
CADRL	0.10	0.00	15.80
LSTM-RL	0.71	0.06	13.01
SARL	0.79	0.16	12.45
P-RL (Our)	**0.90**	**0.05**	**12.21**

6 Conclusion

In this work, we present the P-RL algorithm to dealing with the problem of crowd navigation. The algorithm replaces deep reinforcement learning the MDP model with the POMDP model, and introduces the LSTM neural network into the deep reinforcement learning algorithm. The LSTM neural network has the ability to process time series information to overcome the short-sighted problem of robot trajectory time, and enhances the robustness of the algorithm. And, add the attention network into the neural network, infer the relative importance of the adjacent frame data relative to its future state through the attention network, so as to focus on the key frame data and improve the learning efficiency. Experimental results show our algorithm is preferable to other algorithms in time overhead and navigation success rate in crowds environment.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 961–971 (2016)
2. van den Berg, J.P., Lin, M.C., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: 2008 IEEE International Conference on Robotics and Automation, pp. 1928–1935 (2008)
3. Chen, C., Liu, Y., Kreiss, S., Alahi, A.: Crowd-robot interaction: crowd-aware robot navigation with attention-based deep reinforcement learning. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 6015–6022 (2019)
4. Chen, Y.F., Everett, M., Liu, M., How, J.P.: Socially aware motion planning with deep reinforcement learning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1343–1350 (2017)
5. Chen, Y.F., Liu, M., Everett, M., How, J.P.: Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 285–292 (2017)
6. Ellis, D., Sommerlade, E., Reid, I.D.: Modelling pedestrian trajectory patterns with gaussian processes. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, pp. 1229–1234 (2009)
7. Ferdowsi, A., Challita, U., Saad, W., Mandayam, N.B.: Robust deep reinforcement learning for security and safety in autonomous vehicle systems. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 307–312 (2018)

8. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Phys. Rev. E, Stat. Phys. Plasmas, Fluids Related Interdisc. Top. **51**(5), 4282–4286 (1995)
9. Kayukawa, S., et al.: BBEEP: a sonic collision avoidance system for blind travellers and nearby pedestrians. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (2019)
10. Long, P., Fan, T., Liao, X., Liu, W., Zhang, H., Pan, J.: Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 6252–6259 (2018)
11. Snape, J., van den Berg, J.P., Guy, S.J., Manocha, D.: Smooth and collision-free navigation for multiple robots under differential-drive constraints. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4584–4589 (2010)
12. Sorokin, I., Seleznev, A., Pavlov, M., Fedorov, A., Ignateva, A.: Deep attention recurrent q-network. arXiv:abs/1512.01693 (2015)
13. Vemula, A., Muelling, K., Oh, J.: Social attention: modeling attention in human crowds. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–7 (2018)
14. Wang, J., Meng, M.Q.: Socially compliant path planning for robotic autonomous luggage trolley collection at airports. Sensors (Basel, Switzerland) **19**, 2759 (2019)
15. Zhu, Y., et al.: Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3357–3364 (2017)