Optimizing Pedestrian Simulation Based on Expert Trajectory Guidance and Deep Reinforcement Learning

Senlin Mu
East China Normal University
Xiao Huang
University of Arkansas
Moyang Wang
East China Normal University
Di Zhang
East China Normal University
Dong Xu
Baidu (China)
Xiang Li (xli@geo.ecnu.edu.cn)
East China Normal University

Research Article

Keywords: pedestrian simulation, Deep Reinforcement Learning, Local dilemma, Expert trajectory guidance, Optimization

Posted Date: July 8th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1798752/v1

Optimizing Pedestrian Simulation Based on Expert Trajectory Guidance and Deep Reinforcement Learning

Senlin Mu[1], Xiao Huang[2], Moyang Wang[1], Di Zhang[1], Dong Xu[3], Xiang Li[1,*]

[1] School of Geographic Sciences, East China Normal University, Shanghai 200241, China

[2] Department of Geosciences, University of Arkansas, Fayetteville, Arkansas 72762, United States

[3] Baidu Inc.

[*] Correspondence: xli@geo.ecnu.edu.cn

Abstract

Most traditional pedestrian simulation methods suffer from short-sightedness, as they often choose the best action at the moment without considering the potential congesting situations in the future. To address this issue, we propose a hierarchical model that combines Deep Reinforcement Learning (DRL) and Optimal Reciprocal Velocity Obstacle (ORCA) algorithms to optimize the decision process of pedestrian simulation. For certain complex scenarios prone to local optimality, we include expert trajectory imitation degree in the reward function, aiming to improve pedestrian exploration efficiency by designing simple expert trajectory guidance lines without constructing databases of expert examples and collecting priori datasets. The experimental results show that the proposed method presents great stability and generalizability, evidenced by its capability to adjust the behavioral strategy earlier for the upcoming congestion situations. The overall simulation time for each scenario is reduced by approximately 8%-44% compared to traditional methods. After including the expert trajectory guidance, the convergence speed of the model is greatly improved, evidenced by the reduced 56%-64% simulation time from the first exploration to the global maximum cumulative reward value. The expert trajectory establishes the macro rules while preserving the space for free exploration, avoiding local dilemmas, and achieving optimized training efficiency.

Keywords pedestrian simulation, Deep Reinforcement Learning, Local dilemma, Expert trajectory guidance, Optimization

1 Introduction

With the accelerated urbanization process, various complex buildings and public facilities have started to emerge, with people spending more and more time in these indoor environments. Without a well-developed emergency plan in advance, chaos, crowding, and even trampling accidents could occur when emergencies happen [1]. In recent years, crowd evacuation has been a hot issue in domestic and international research. Due to the disadvantages of costly exercises, it is often difficult to organize evacuation drills in the field [2]. Consequently, the pedestrian simulation technology based on computer simulation has become the main means to study crowd evacuation nowadays. Currently, traditional pedestrian simulation models include the social force model (SFM) [3], cellular automata model (CAM) [4], fluid mechanics model [5], and agent-based model, to list a few. However, SFM fails to obtain smooth motion trajectories due to its necessity to balance the relationship between various virtual forces [6]. The individuals in the CAM are restricted to the grid and can only consider information from the surrounding neighborhoods [7]. As for fluid mechanics models, they focus on describing the overall

motion trends, largely ignoring the interactions between individuals. Pathfinder software uses the agent-based model, which is a widely used evacuation simulation software. However, unreasonable moving behaviors tend to occur in many agent-based simulations. Therefore, it is crucial to develop novel and improved pedestrian simulation models.

In recent years, as Deep Reinforcement Learning (DRL) has made significant breakthroughs in video games [8], robot navigation [9], and recommendation systems [10], scholars started to apply DRL to pedestrian simulation. DRL integrates the powerful feature representation capability of deep learning (DL) and the excellent decision-making capability of reinforcement learning (RL), realizing the self-supervised learning of agents and completing the decision-making in a high-dimensional state and action space. Yao [11] proposed a method based on RL and a deep residual network to simulate crowd motion. Yao's method boosted the realism of the simulation but with low evacuation efficiency. Xu et al. [12] proposed a hierarchical model consisting of Proximal Policy Optimization (PPO) and Optimal Reciprocal Velocity Obstacle (ORCA) for pedestrian simulation in local space, considering global path smoothing and local collision avoidance. The study by Xu et al. [12] used virtual visual rays to obtain the external environment. However, the computational complexity of this method increases exponentially with the number of rays. Some scholars [13,14] improved the multi-agent DRL method and achieved stable and effective strategies in some competitive and cooperative scenarios. However, these multi-agent RL algorithms generate a huge state space with the increase in the number of agents, leading to the curse of dimensionality. In addition, given the fact that DRL methods need to interact with the environment to obtain training data, The training time for DRL models is considerably long. In complex scenes containing dead ends and promenades, agents are very likely to fall into the dilemma of local optimality [15]. Therefore, it is crucial to explore effective optimizing means to reduce the learning difficulty of agents and improve the model training efficiency while outputting reasonable and feasible simulation results.

In this study, we propose a single-agent hierarchical pedestrian simulation model that combines DRL and local collision avoidance, named D3QN-ORCA. To mitigate the issue of falling into local optimum in certain complex scenes with dead ends and promenades, we introduce expert trajectory guidance. By adding the expert trajectory imitation degree to the reward function, our agents are guided to avoid local dilemmas, leading to improved training efficiency.

2 Related works

2.1 Traditional simulation model

In terms of the differences in spatial perspectives, traditional pedestrian simulation models can be divided into macroscopic models and microscopic models. The macroscopic approach mainly considers the global path planning problem, represented by the fluid mechanics model and the model based on the potential energy field. Yang [16] proposed an improved hydrodynamic model of pedestrian flow, obtained evacuation characteristics in several typical evacuation scenarios, and achieved pedestrian flow self-organization. Bounini et al. [17] searched for feasible paths in the potential field according to the potential gradient descent algorithm, added a repulsive potential to the current state in the blockage case, and overcame the local minimal problem of traditional potential methods; Wu [18] introduced the concept of the dynamic potential energy of the grid, which guides the movement of the crowd based on the crowd density on the dynamic division of the potential energy grid.

In contrast, microscopic models, such as SFM, CAM, and ORCA, focus more on the interaction and

local control among individuals. Zhao [7] proposed an adaptive method to calculate the optimal motion vector of pedestrians. He improved the expected speed and direction derived from the pedestrian self-driven force in SFM and enhanced the realism of crowd evacuation; Ma et al. [21] introduced the active avoidance force in SFM and combined it with the contact theory of the discrete element model of particles. The model improves the irrationality of the avoidance behavior of pedestrians walking close to each other in the original SFM simulation.

ORCA solves the avoidance jittering behavior of the velocity obstacle (VO) model and the collision avoidance dilemma of multiple agents, transforming the velocity selection into a simple linear programming problem. Guo et al. [20] proposed the VR-ORCA approach that abandoned the assumption that a pair of agents take half of the collision avoidance responsibility in the original ORCA and only required their responsibility to sum to one. This study solves the asymmetric situation faced by neighboring agents and reduces the probability of pedestrian collision and passage time. He et al. [21] combined shadow obstacles with ORCA for large-scale crowd evacuation analysis. Compared with the SFM model, it produces simulation with great realism with high computational efficiency. In view of the superiority and efficiency of ORCA, we use it as the underlying collision avoidance mechanism for pedestrian simulation as a way to control the interaction between individual pedestrians.

2.2 Deep reinforcement learning

With the launch of AlphaGo [22] by the DeepMind team, which defeated the human Go world champion, DRL began to receive widespread attention. DRL combines the neural network perception capability of DL and the interactive trial-and-error idea of RL to realize the decision-making process in a high-dimensional state and action space. Deep Q Network (DQN) [23], the first DRL algorithm, is a common algorithm applied in discrete action space scenarios. The DeepMind team fed original game images from Atari 2600 into a convolutional neural network and used tricks like experience replay and target network to achieve results beyond the level of top human players in dozens of games. Since then, DQN-based variants have started to emerge. For example, to solve the overestimation problem of DQN, the Double Deep Q Network (DDQN) algorithm uses a dual network structure for action selection and value evaluation, respectively [24]; the Dueling Double Deep Q Network (D3QN) algorithm improves the stability of the algorithm and the accuracy of action selection by improving the neural network structure, decomposing the network into a state value function network and an advantage function network [25]; The Prioritized Experience Replay DQN algorithm [26] uses Temporal Difference (TD) error to measure the importance level of experience trajectories, introduces random priority sampling, importance sampling, among others, to improve the slow training problem in reward-sparse environments.

In continuous action space scenarios, policy gradient methods are more applicable, such as the REINFORCE algorithm with reduced variance with baseline [27] and trust region policy optimization (TRPO) [28], which mitigates difficulty in determining the learning step size and proximal policy optimization (PPO) [29] algorithm. Despite their slightly better performance than the DQN series algorithms in terms of convergence and stability, they own notable disadvantages: easy to fall into local optimum, large trajectory variances, and low sample utilization.

Scholars have combined value-based methods with policy-based methods and proposed the actor-critic (AC) method. The actor updates the action based on the policy, while the critic evaluates the action through the value function. Some representative algorithms include the deep deterministic policy gradient (DDPG) [30], twin delayed deep deterministic policy gradient (TD3) [31], and the soft actor-critic algorithm (SAC) [32], to list a few. Although the above algorithms integrate the advantages of

value-based and policy-based methods, they are hyperparameter-sensitive.

2.3 DRL-based pedestrian simulation and optimization

As pedestrian simulation can be modeled as a Markov decision processes (MDP) problem and RL considers MDP to find the optimal policy and maximize the expected total return, a number of studies have applied RL to the field of pedestrian simulation. Lee et al. [33] proposed a crowd simulation method based on the AC framework. By setting a simple reward function, their agents are able to perceive the surrounding environment and the situation of neighboring agents and make decisions independently to achieve collision avoidance and end-point approaching. Xu et al. [12] proposed a hierarchical simulation model combining PPO and ORCA, using ray perception of virtual vision as the state input to obtain the optimal policy for the movement of the agents. They verified the superiority of the algorithm in several classic scenarios. Sharma [34] et al. pre-trained the network weights of DQN to incorporate the shortest path information and added the importance vector to the action output, leading to significantly simplified action space with reduced training time.

Expert knowledge assistance is an effective means to optimize problems prone to occur in RL, such as ineffective exploration and local optimality. Many scholars used expert strategies in the form of Behavior Cloning (BC) to guide training. Although BC omits the time to interact with the environment, it is prone to error accumulation. In other words, agents may fail to fit the expert behavior correctly at some point, step into an unfamiliar state that does not exist in the a priori data set, and continues to make decisions that deviate from the expert trajectory, eventually leading to data drift [35]. Offline RL [36] also uses static datasets to obtain the best strategy. Unlike BC, which is essentially a form of supervised learning, offline RL is still based on standard RL algorithms. All of the above expert experience-guided methods rely heavily on a priori data, and the quality and composition of the a priori data greatly affect the quality and efficiency of agent learning.

Therefore, for complex environments containing promenades and dead ends, the dilemma of local optimum still exists in current pedestrian simulation research. In this study, when planning the global path of the agent with DRL, we supplement expert knowledge, thicken the sparse reward, and introduce expert trajectory guidance into the reward function. The proposed approach gives the pedestrian macro-rule guidance while encouraging independent exploration, leading to enhanced exploration efficiency.

3 Methodology

3.1 Methodology overview

In this study, we define the simulation environment as a two-dimensional plane space. The action decision interval is set to one time step, i.e., one frame f, the total number of simulation frames is F_{max}, and the total number of round frames is M. Each pedestrian i can be described as a circle with attributes $\alpha_i = \{r_i, p_i, v_i^{pre}, v_i^{fin}\}$, where r_i denotes the pedestrian radius, p_i denotes the current position of the pedestrian, v_i^{pre} represents the initial desired speed of the pedestrian, and v_i^{fin} corresponds to the final speed of the pedestrian after collision avoidance adjustment. The task of pedestrians is to reach the specified target point G_i with the shortest time in each round while trying to avoid collision with other agents.

The whole simulation process can be divided into two parts, i.e., environment interaction sampling and neural network training (as shown in Fig. 1). The specific steps of environmental interaction sampling in Fig. 1(a) are as follows.

(1) Obtain the state s_t according to the current environment.

(2) Input the state s_t into the neural network, obtain the expected value of each action after noise interference, and output the action a_t with the largest value.

(3) Transform the output action a_t into a velocity vector v_t^{pre} and send it to the ORCA model for velocity adjustment, i.e., obtaining collision-free velocity v_t^{fin}.

(4) Update the pedestrian position according to v_t^{fin}, obtain the reward r_t and the next state s_{t+1}.

(5) Deposit the experience data $\{ s_t, a_t, r_t, s_{t+1} \}$ into the experience replay pool. If the amount of data in the experience pool has reached the maximum capacity value $E_{capacity}$, the old data are eliminated, with new data deposited in order; otherwise, the new data are directly deposited.

(6) Repeat steps (1)-(4) until the number of round steps reaches M or all pedestrians reach the target point. Then, this round simulation ends, and the simulation environment is reset.

(7) When the total number of simulation steps reaches F_{max}, the environment interaction process is terminated.

The specific steps of the neural network training part in Fig. 1(b) are as follows:

(1) Take $Batch\_Size$ bars of experience data from the experience replay pool and feed them into the neural network.

(2) Calculate the loss function to train the policy network θ_{cur} based on the policy network θ_{cur} and the target network θ_{tar}.

(3) Copy the network parameters of the policy network θ_{cur} to the target network θ_{tar} at every fixed number of f_{copy} steps.

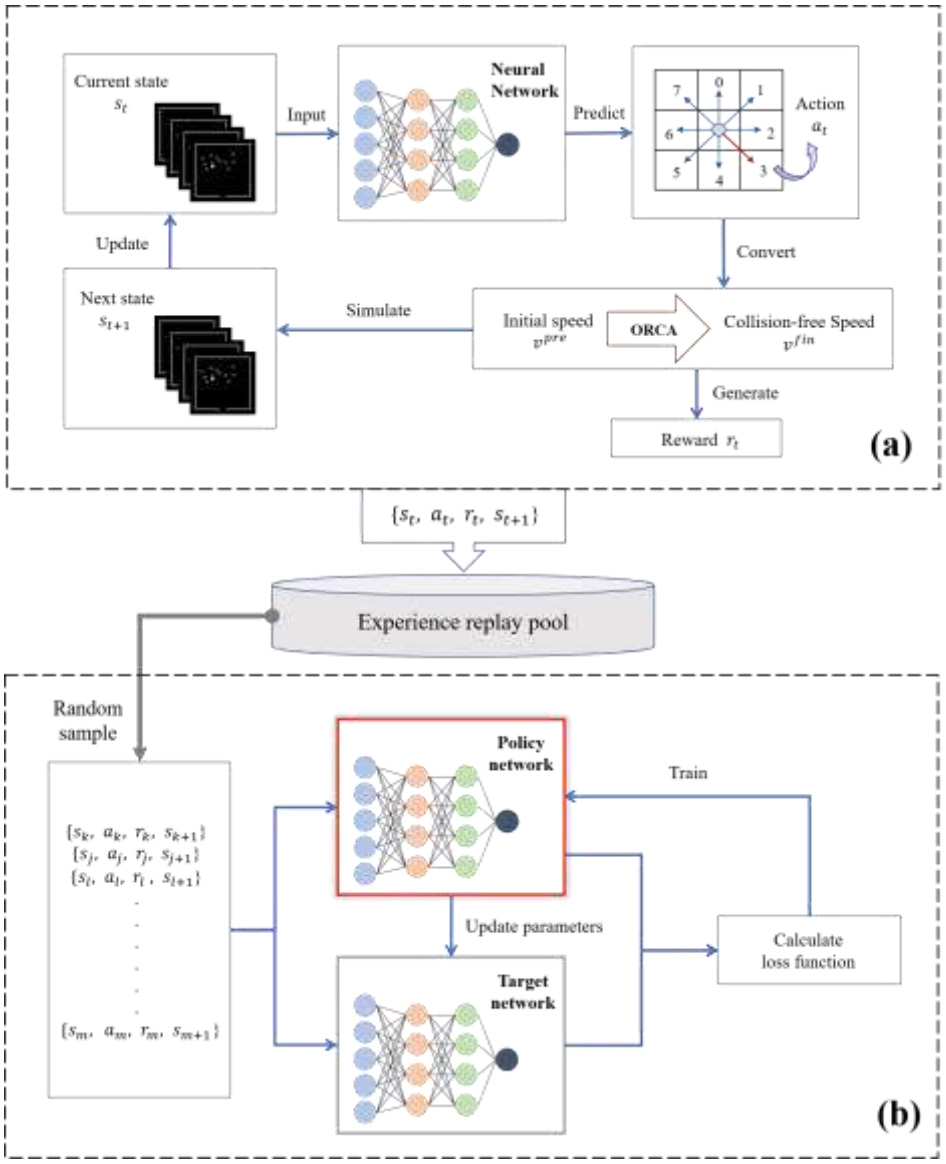


Fig.1 The methodology framework of the proposed approach.

3.2 Deep reinforcement learning model for pedestrian simulation

3.2.1 State space

In the same way as the input of the Atari game [23], the state space in this study is designed as scene images. To reduce the state input dimension, we convert the color image into a grayscale map and scale the image size to 84*84. Meanwhile, the current frame is superimposed with four consecutive images of the previous three frames as the current state, i.e., $s_t = \{f_{t-3}, f_{t-2}, f_{t-1}. f_t\}$, to show the movement trend of the pedestrians. In addition, to distinguish the current decision pedestrian from other pedestrians, the pixel value of the decision pedestrian is set to 255, the pixel value of other pedestrians and obstacles is set to 100 with the same pixel value, and the background pixel value is set to 0 (as shown in Fig. 2).

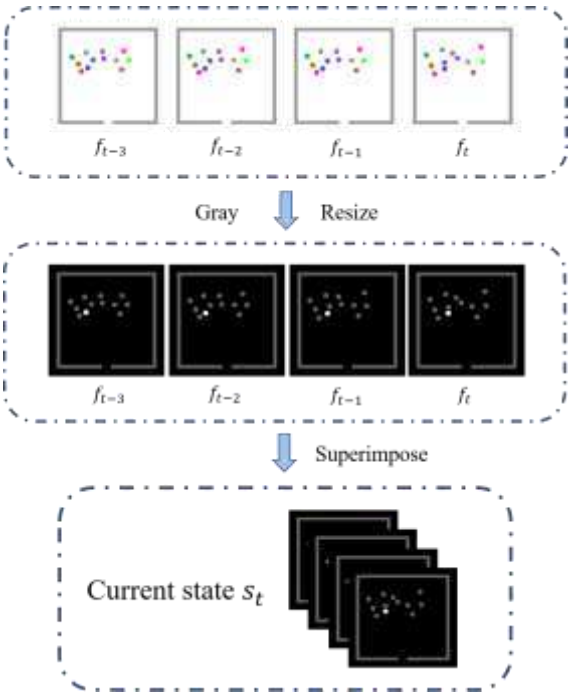


Fig.2 State space.

3.2.2 Action space

The initial desired velocity v^{pre} of the pedestrian can be decomposed into the velocity direction v_θ and the velocity magnitude v_μ. Here, we assume that v_μ is always the maximum velocity limit v_{max} of the pedestrian, and the action to be obtained is v_θ. To reduce the action space and retain certain accuracy, we discretize the action space $\mathbb{Z}$ of pedestrians into eight directions. According to the general orientation method, a direction is divided every 45 degrees interval starting from due north, and index values 0-7 are assigned in clockwise order. As a result, the action space $\mathbb{Z}$ of pedestrian i can be expressed as $\mathbb{Z} = [a_i^k]$, and the action decision variables $a_i^k = \{0,1,2,\ldots,7\}$, representing the eight directions of up, right up, right, right down, down, left down, left, and left up (as shown in Fig. 3). Further, the predicted action direction is transformed into the velocity direction v_θ, which are combined with the velocity magnitude v_μ to output the initial desired velocity v^{pre}. Finally, the final collision-free velocity v^{fin} is further calculated from ORCA.

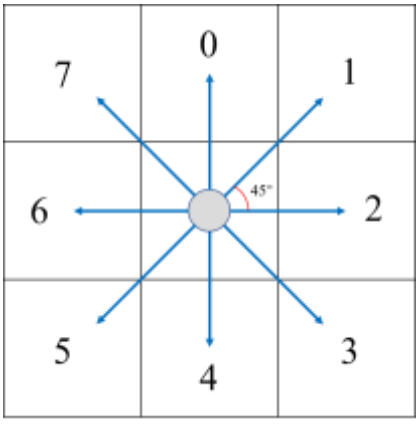


Fig.3 The action space of an agent in this study.

3.2.3 Reward function

RL quantitatively evaluates decision actions by means of a reward function, and its optimization goal is to maximize the cumulative reward to guide an agent to explore the global optimal solution. Therefore, the design of a reasonable reward function is crucial for the convergence and stability of RL. Given that the pedestrian simulation aims to explore globally optimal paths and avoid collisions, we perform reward shaping for sparse rewards. Meanwhile, we introduce an expert trajectory simulation reward to enhance exploration efficiency and optimize simulation results for some complex environments that are prone to local dilemmas.

The reward function R defined in this study consists of four components, i.e., goal reward r_{goal}, collision avoidance reward $r_{collision}$, movement reward r_{punish}, and expert trajectory imitation degree reward $r_{imitation}$. The composition form of the reward function is distinguished in different scenarios $E = \{e_{simple}, e_{complex}\}$ (e_{simple} represents a simple environment, $e_{complex}$ represents complex environment).

$$R = \begin{cases} \omega_1 r_{goal} + \omega_2 r_{collision} + \omega_3 r_{punish} & E = e_{simple} \\ \omega_1 r_{goal} + \omega_2 r_{collision} + \omega_3 r_{punish} + \omega_4 r_{imitation} & E = e_{complex} \end{cases} \tag{1}$$

where $\omega_1, \omega_2, \omega_3, \omega_4$ are the weight coefficients. The target reward r_{goal} is used to guide pedestrians to approach the target point in the following form:

$$r_{goal} = \begin{cases} dis(p_i^{t-1}, G_i) - dis(p_i^t, G_i) & dis(p^t, G_i) > D_{threshold} \\ w_a & dis(p^t, G_i) \le D_{t} \end{cases} \tag{2}$$

where w is the end-point reward value for reaching the target point, $D_{threshold}$ is the distance determination threshold. The current pedestrian reaches the end point when the distance between the agent and the target point is less than the threshold. $dis(\ )$ denotes the Euclidean distance measure. p_i^t、p_i^{t-1} denote the positions of pedestrian i at moments t and $t-1$, respectively. Before the pedestrian reaches the target point, agents are encouraged to move closer to the target point through the difference form in formula (2), thickening the reward distribution in the exploration process.

The collision avoidance reward $r_{collision}$ is used to avoid mutual collisions between pedestrians and pedestrians and obstacles. It can be defined as:

$$r_{collision} = \frac{v^{pre} \cdot v^{fin}}{|v^{pre}| \cdot |v^{fin}|} \tag{3}$$

where the denominator is the dot product of the initial desired velocity v^{pre} and the final collision-free velocity v^{fin}. The numerator is the product of the respective mode lengths, which represent the degree of similarity between the two vectors and map the values to the range [-1,1].

The movement reward r_{punish} is used to control the number of steps the pedestrian moves and can be written in the following form:

$$r_{punish} = w_b \tag{4}$$

where w_b is the hyperparameter generally set to a fixed negative value to encourage pedestrians to reach the target point with the least number of steps.

For some complex environments, agents often fail to explore the optimal strategy, leading to slow convergence. Thus, we add expert trajectory guidance in some locations of the scene. Suppose the current position of pedestrian i is p_i^t and its predefined expert trajectory is $\mathcal{L}_i = \{T_i^1, T_i^2, \dots, T_i^n\}$, T_i^k is the node of the expert trajectory path, $T_i^k \to T_i^{k+1}$ represents the road section from node k to node $k+1$. We calculate the distance between p_i^t and each road section at this time, find the closest road section $T_i^t \to T_i^{t+1}$, thus obtaining the trajectory direction of the road section $d_{guide} = T_i^{t+1} - T_i^t$. The expert

trajectory imitation degree reward $r_{imitation}$ can be defined as:

$$r_{imitation} = \begin{cases} \dfrac{v^{pre} \cdot d_{guide}}{|v^{pre}| \cdot |d_{guide}|} + w_c * e^{-dis(p_i^t \ , \ T_i^t \to T_i^{t+1})} & t+1 \le n \\ w_d \end{cases} \quad (5)$$

where the similarity between the predicted velocity v^{pre} and the trajectory direction of the current road section is described in the form of a vector dot product. The distance between the current position p_i^t to the current imitated road section $T_i^t \to T_i^{t+1}$ is described by $e^{-dis(p_i^t \ , \ T_i^t \to T_i^{t+1})}$. w_c is a weighting factor that aims to balance the direction simulation degree and distance excursion. w_d is the fixed reward value given to the agent after it has simulated $\mathcal{L}_i$.

3.3 Hierarchical model based on D3QN and ORCA

3.3.1 ORCA algorithm

For obstacle avoidance, we implement ORCA for short-range obstacle avoidance control. ORCA solves the problem of frequent jitter and mitigates the difficulty in multi-agent planning in the VO algorithm. It transforms the velocity space into a bipartite plane and finds the optimal solution using simple linear programming to achieve effective obstacle avoidance for dense crowds [37].

Suppose there exists a pedestrian $\alpha_A = \{r_A, p_A, v_A^{pre}, v_A^{fin}\}$ and a pedestrian $\alpha_B = \{r_B, p_B, v_B^{pre}, v_B^{fin}\}$, we define V_{A-B} to be the initial relative desired velocity (v_A^{pre}、v_B^{pre} pointing to the target point respectively) and the initial velocity collision range to be a circle D with center P and radius R.

$$D(P,R) = \{V_{A-B} | \ \|V_{A-B} - P\| < R\} \quad (6)$$

Considering the continuity of the moving process, the velocity barrier region is extended as a truncated cone, including circle D and its rear range (as shown in the shaded part of Fig. 4). Thus, the velocity barrier region $VO_{A|B}$ of pedestrian A relative to pedestrian B can be described as:

$$VO_{A|B} = \{v \exists t \in [0,\tau] :: tv \in D(p_B - p_A, r_A + r_B) \quad (7)$$

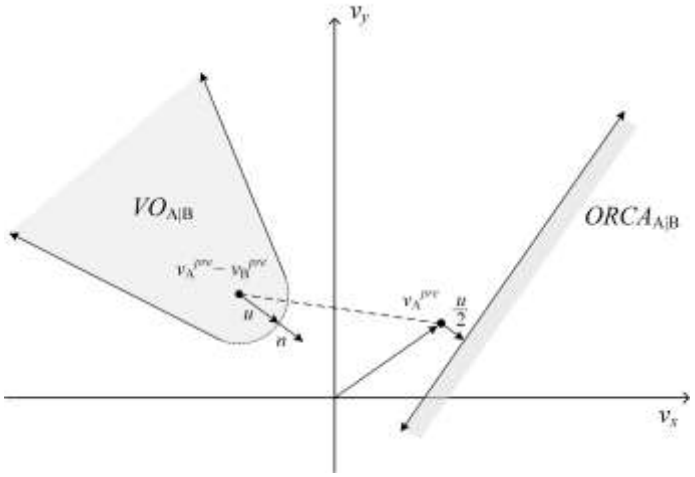


Fig.4 The VO region and ORCA half-plane.

The collision will occur in time τ, if relative velocity falls in $VO_{A|B}$. At this point, it is necessary to find a collision-avoidance vector u, which is the shortest vector from the relative expected velocity V_{A-B} to the boundary of $VO_{A|B}$. The normal of u to the regional boundary position as n. Therefore, based on the principle of mutual avoidance, A only needs to change $\frac{1}{2}u$ of the velocity while the direction is

the half-plane pointed by n. The half-plane $ORCA_{A|B}$ can be expressed as:

$$ORCA_{A|B} = \{v \mid \left(v - v_A^{pre} + \frac{1}{2}u\right) \cdot n \geq 0\} \tag{8}$$

We calculate the ORCA half-planes of A and other agents in turn and generate the half-plane intersection $ORCA_A$. If v_A^{pre} falls within $ORCA_A$, the final collision-free velocity v_A^{fin} of A is v_A^{pre}; otherwise, the velocity closest to v_A^{pre} is taken in the intersection set. For scenarios with a dense crowd, there exists a situation where the half-plane intersection set is the empty set. We choose the shortest velocity of length from the current velocity point to each half-plane Euclidean distance maximizing velocity (as shown in Fig. 5).

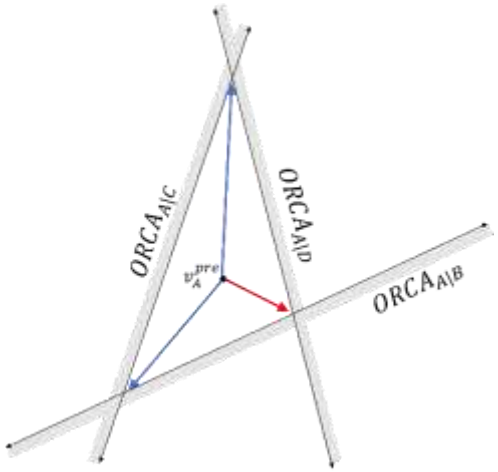


Fig.5 Speed selection when a half-plane intersection set is the empty set.

3.3.2 D3QN algorithm

In general, the state of the pedestrian at the next moment is only related to the current state, which means that the pedestrian motion process owns the Markovian property. Therefore, global path planning at the top level can be achieved by DRL. Since we use the scene picture as the state space input, the current decision pedestrian can perceive the motion state of the rest of the pedestrians, so we use the single-agent-based D3QN algorithm that does not need to consider individual collaboration.

The core idea of the DQN algorithms is to use the value function $Q^{\pi}(s_t, a_t; \theta)$ to evaluate the value of executing action a_t for state s_t under policy π predicted by a neural network with parameter θ[23]. Q^{π} is known to satisfy the Bellman equation:

$$Q^{\pi}(s_t, a_t; \theta) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta) \tag{9}$$

To optimize Q^{π}, the mean square error loss function in the neural network is defined based on TD error as follows:

$$L(\theta) = (r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_{tar}) - Q(s_t, a_t; \theta_{cur}))^2 \tag{10}$$

Here, two identical networks are introduced to solve the training instability problem. The policy network θ_{cur} updates the network parameters in real-time to calculate the predicted values, while the target network θ_{tar} is relatively fixed to calculate the target values, and the parameters are copied from θ_{cur} every f_{copy} step.

By decoupling action selection and value calculation, Double DQN uses the policy network θ_{cur} to select the action and brings in the target network θ_{tar} to determine the action value. It mitigates the

overestimation problem that tends to occur in the basic DQN to a certain extent [24], and only needs to replace the target value calculation function as follows:

$$Q^{\pi}(s_t, a_t; \theta) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, argmax_a Q(s_{t+1}, a_{t+1}; \theta_{cur}); \theta_{tar}) \tag{11}$$

Further, Dueling DQN changes the original network structure of DQN and proposes the concept of the dyadic network by splitting the original output value function $Q^{\pi}(s_t, a_t; \theta)$ into two branches, i.e., the state value function $V(s_t; \theta^{\alpha})$ and the action advantage function $A(s_t, a_t; \theta^{\beta})$ [25]. The state value function is used to predict the goodness of the state, while the action advantage function is used to predict the importance of each action under the state s_t:

$$Q^{\pi}(s_t, a_t; \theta) = V(s_t; \theta^{\alpha}) + A(s_t, a_t; \theta^{\beta}) \tag{12}$$

At the same time, for its "unidentifiable" problem, it is necessary to set the output vector sum of the action advantage function to 0. The value function can be rewritten as follows:

$$Q^{\pi}(s_t, a_t; \theta) = V(s_t; \theta^{\alpha}) + A(s_t, a_t; \theta^{\beta}) - mean_a A(s_t, a_t; \theta^{\beta}) \tag{13}$$

Combined with the actual problem of pedestrian simulation, in many cases, the size of the value function Q is often not related to the action but influenced by the environment. For example, when the scene space and target location are sufficiently empty, the left-right movement of the agent at this moment may have no effect on the result, while the choice of action only becomes particularly important when there is an obstacle in front. Therefore, splitting the value function is an effective means to improve prediction accuracy.

In addition, to improve the exploration efficiency, we replace the traditional $\varepsilon - greedy$ exploration method with Noisy net [38]. Compared with adding noises to the action at the output of the network, adding noises to the network parameters seems more reasonable and effective. Given a linear cell with input y and output x:

$$y = w \cdot x + b \tag{14}$$

where w is the weight and b is the bias, noise ε^w、ε^b are sampled from the independent Gaussian distribution. The noise linear unit can be described as:

$$y = (\mu^w + \sigma^w \odot \varepsilon^w) \cdot x + \mu^b + \sigma^b \odot \varepsilon^b \tag{15}$$

where $\odot$ is noted in dot product form, μ and σ represent the mean and variance, respectively.

The network structure of the target network and the policy network is shown in Fig. 6. The state space is 84 × 84 × 4, and the input states pass through three convolutional layers with convolutional kernel sizes of 8 × 8 × 4, 4 × 4 × 32, and 3 × 3 × 64, respectively. Further, abstract features extracted from the convolutional layers are fed to two fully connected layer branches with noise. One branch represents the scalar state value V and the other branch represents the vector action advantage function A, with the vector length being the action space length. Finally, the results of the two branches are aggregated and output as each action value Q.

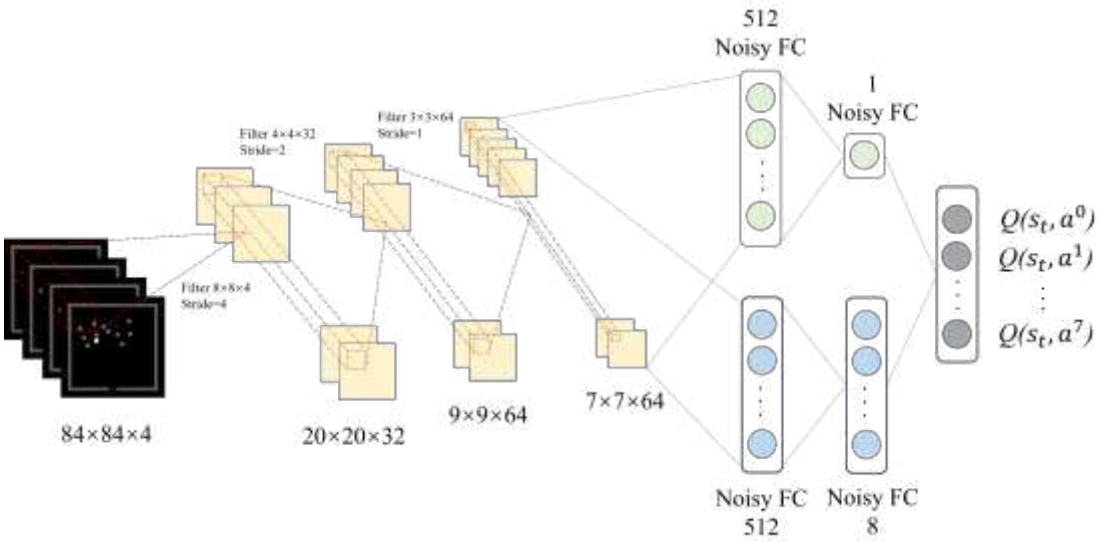


Fig.6 The neural network structure proposed in this study.

4 Experiments and Results

To verify the effectiveness and superiority of the proposed method, we divide the experiment into two parts and evaluate them in two scenarios, respectively.

- In the simple environment, we compare the simulation results of Pathfinder software with the traditional simulation model and the proposed D3QN-ORCA hierarchical model. The results show that the proposed method is able to achieve higher quality pedestrian simulation, taking into account the local collision avoidance among individuals while planning the global path.
- In the complex environment, we compare the simulation results of Pathfinder software and D3QN-ORCA with the addition of expert trajectory guidance. The results prove the generalization of the proposed method for complex scenarios that are prone to local optimums. The addition of expert trajectory can reduce useless exploration and effectively guide the optimal path. The results demonstrate that the proposed method presents fast and stable convergence with the change of global reward value.

Table 1 shows some of the hyperparameters covered in this study.

Table 1 Hyperparameter settings.

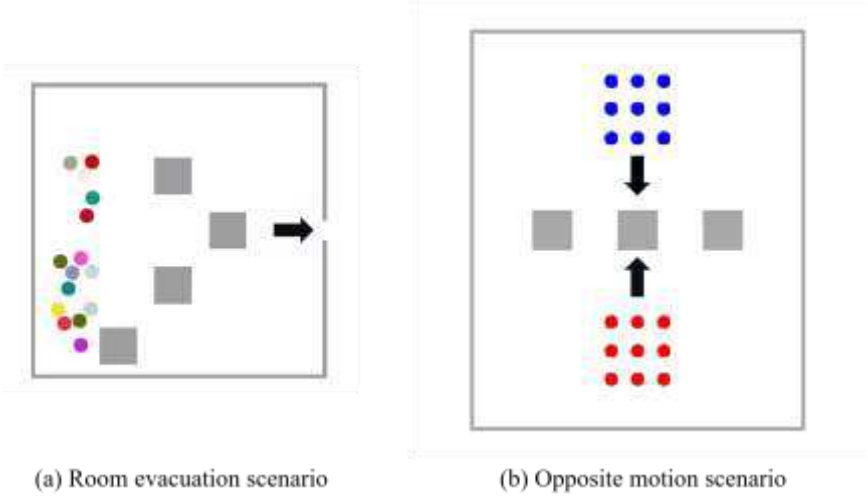
Hyperparameter	Value
Learning rate	1e-4
Discount factor γ	0.99
Experience replay capacity $E_{capacity}$	7e-4
$Batch\_Size$	256
Total number of simulation steps F_{max}	6e-5
Parameter update interval f_{copy}	1e-3
The maximum speed limit for pedestrians v_{max}	2
Pedestrian radius r	2
Distance Threshold $D_{threshold}$	1
ω_1	1.25
ω_2	1.25

ω_3	1.0
ω_4	1.0
w_b	2.1
w_c	1.25
w_d	1.25

4.1 Simple environment

The simple environment aims to verify that combining DRL and ORCA is able to complement each other and achieve a more refined simulation of pedestrian behavior than traditional models. Therefore, we design two experimental scenarios:

- **Room evacuation scenario**. The single exit room evacuation scenario is often used to evaluate the behavioral strategy of crowd evacuation. In this scenario, we visualize how pedestrians in a crowded and chaotic state make decisions to reach the exit as soon as possible. We set the environment as a square room with a length/width of 80 and the exit width set to 5, allowing only one person to pass through. We also place four static obstacles in the room to increase the complexity of the environment (as shown in Fig.7(a)).
- **Opposite motion scenario**. The opposite direction movement is a classic scenario in pedestrian simulation. It shows the behavior of two groups of pedestrians moving close to each other face to face in an open environment and crossing each other to reach the end point on the other side. We set the road length to 120 and the width to 100 and place three static obstacles between the two groups of pedestrians to increase the difficulty of pedestrian decision-making (as shown in Fig.7(b)).



(a) Room evacuation scenario (b) Opposite motion scenario

Fig.7 Simple environment

4.1.1 Room evacuation scenario

In this scenario, we generate 15 random pedestrians on the left side of the room, whose goal is to exit the room using the right-side door. We visualize the current crowd state every 20 frames to explore pedestrians' behaviors. In particular, the differences between our model and Pathfinder software in terms of simulated pedestrian movements between obstacles and at exits are compared by five consecutive screenshots during congestion.

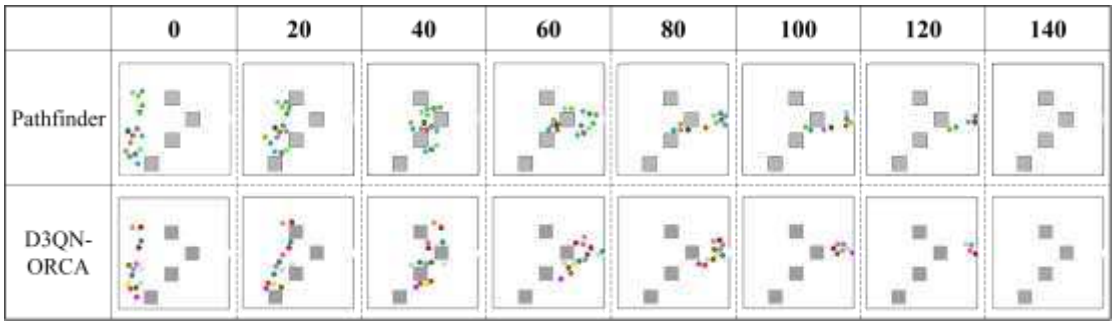
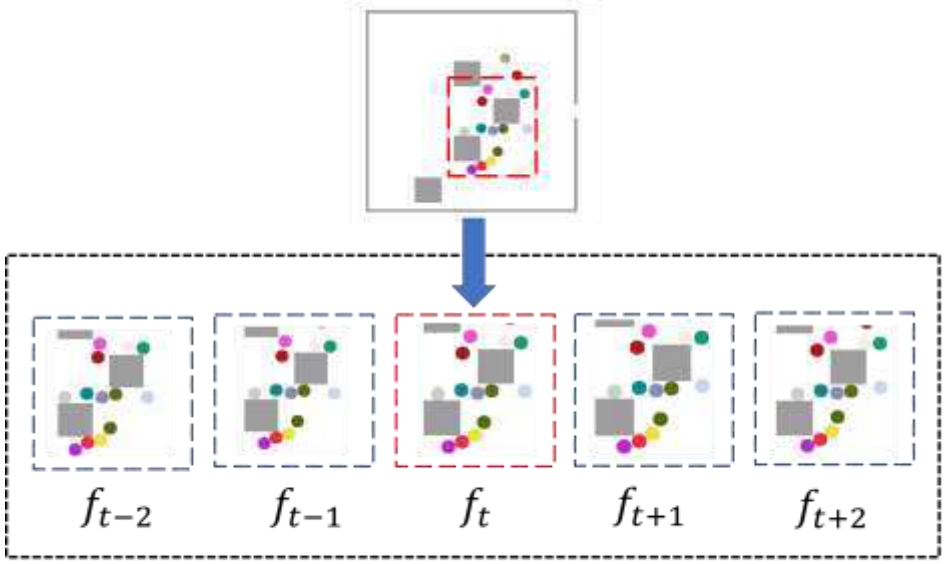
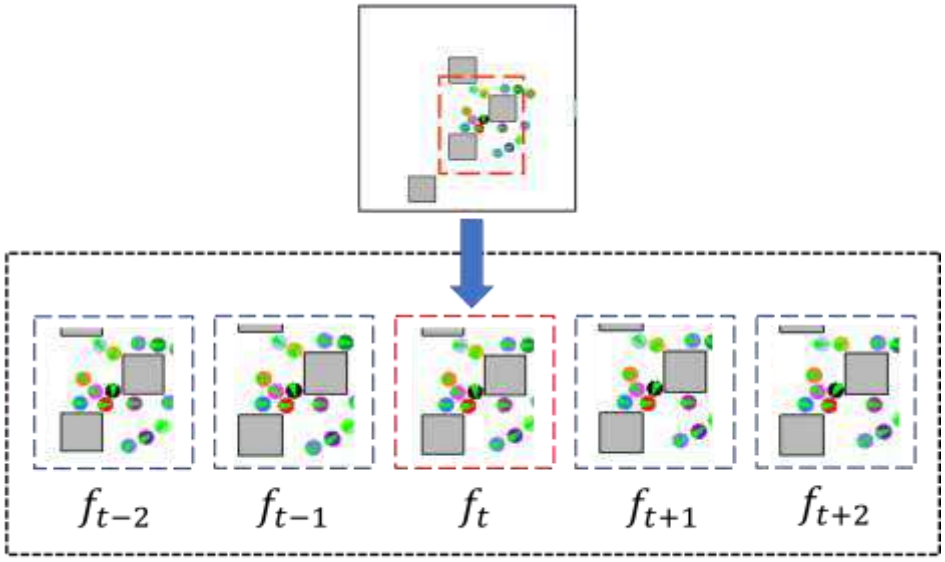


Fig.8 Simulation screenshots of room evacuation scenario (Pathfinder took a total of 138 frames, while the proposed D3QN-ORCA took a total of 127 frames)

We notice that both models (i.e., Pathfinder and the proposed D3QN-ORCA) eventually complete crowd evacuation by frame 140, but their agents behave differently when approaching obstacles and at the exit (Fig.8). Until frame 20, both groups of pedestrians maintain the same movement strategy, but after that, attracted by the target point, the pedestrians in Pathfinder start to converge toward the middle of the scene. At frame 40, two pedestrians in D3QN-ORCA choose to move towards the top obstacle, while the pedestrians in the middle of the scene have adjusted their position and formed an orderly queue to pass through the passage between the obstacles to avoid congestion. Agents in Pathfinder create a blockage, with a few pedestrians still stranded between the barrier aisles until frame 60. The same situation occurs at the exit. In comparison, pedestrians in D3QN-ORCA try to reduce the evacuation time by passing the exit one by one, as the width of the exit is only for one person. We notice that Pathfinder fails to present the ability to maximize the global gain, evidenced by the congestion at the exit. Eventually, pedestrians in Pathfinder take 138 frames to complete the evacuation, while the ones in D3QN-ORCA take only 127 frames.



(a) D3QN-ORCA



(b) Pathfinder

Fig.9 Five consecutive frames that present the motion process when passing through the passages between obstacles in Pathfinder and the proposed D3QN-ORCA methods.

Fig.9 presents the motion process of the crowd simulated by the two methods in passing through the passage between obstacles for five consecutive frames, namely [f_{t-2} ,f_{t-1} ,f_t ,f_{t-1} ,f_{t+2}]. It can be seen that pedestrians in D3QN-ORCA change the original travel route of pedestrians in advance before entering the narrow passage, and the crowd forms a neat and orderly queue and crosses the obstacles in turn. In comparison, congestion occurs for pedestrians in Pathfinder in the aisle. The great performance of the proposed D3QN-ORCA is due to its capability to maximize the overall reward by guiding the pedestrians through the DRL while avoiding collisions and optimizing the global path, while the Pathfinder makes decisions only for the current moment without considering subsequent situations.

4.1.2 Opposite motion scenario

To analyze the behavior of pedestrians moving in opposite directions, we generate nine pedestrians on each side of the obstacles, whose goals are to reach the other side of the scene boundary. In this case, we focus on the intersection of the pedestrian streams every ten frames, followed by an interval of 30 frames to show the difference in the time taken to reach the target point (Fig.10).

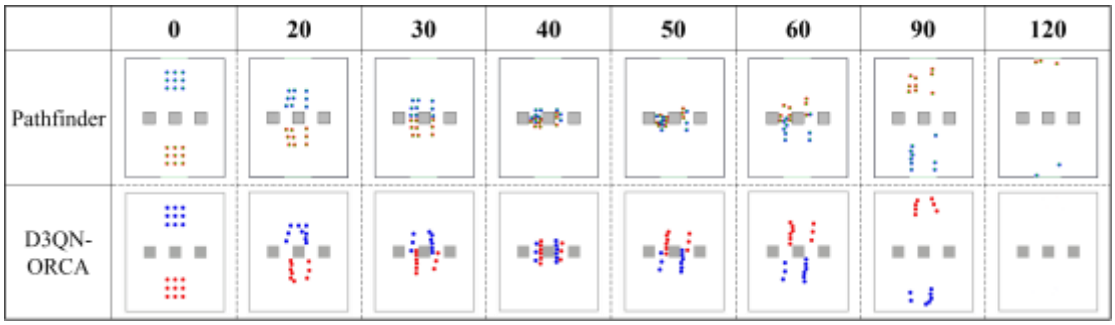


Fig.10 Selected frames in opposite motion scenario (Pathfinder took a total of 129 frames, and D3QN-ORCA took a total of 112 frames)

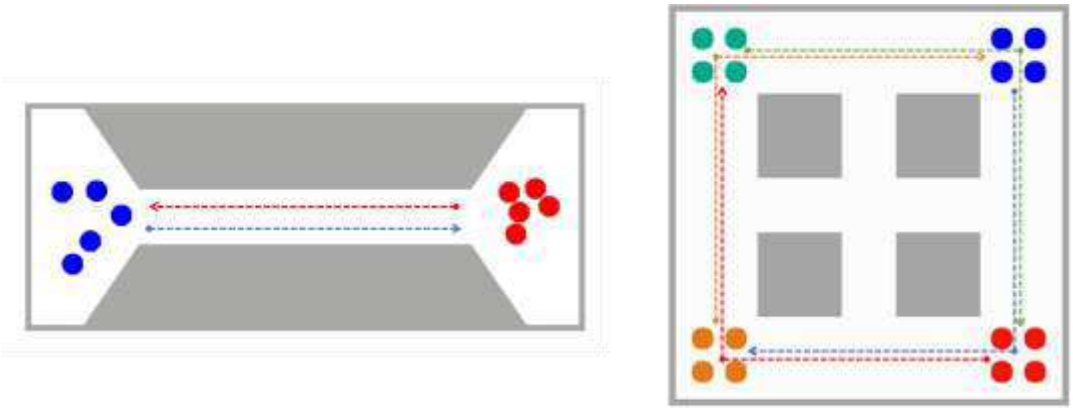
Within the first 20 frames, pedestrians in both approaches avoid static obstacles by splitting the queue in two. However, as the two groups of pedestrians are about to meet, their behavior patterns start to differ. We notice that, at frame 30, two groups of pedestrians in D3QN-ORCA have already transformed into an orderly queue to ensure that they do not collide with the oncoming crowd during the subsequent movement. At frame 60, two groups of pedestrians in D3QN-ORCA have completed the intersection of the crowd with a neat formation. In contrast, Pathfinder still plans the shortest path to the target point for all pedestrians when they meet in close proximity and are about to collide, without taking into account the future congested situation that may occur. The pedestrians in Pathfinder spend a considerable amount of time getting out of the local dilemma in frames 40-60. Eventually, five pedestrians are stranded in the scene until frame 120, while D3QN-ORCA completes the simulation in frame 112. The above observations demonstrate that, due to the powerful autonomous exploration capability of DRL, pedestrians are able to continuously adjust their own strategies by interacting with the environment. At the same time, DRL combined with ORCA achieves inter-individual collision avoidance, thus maximizing the global cumulative reward and optimizing the pedestrians' paths.

4.2 Complex environment

In this section, we further complicate the scenario based on the simple environment to explore the generalization of the proposed method in this study and the effectiveness of adding expert trajectory guidance in some extreme environments where it is easy to fall into local optima. We design two scenarios:

- **Narrow promenade encounter scenario**. In general, conventional pedestrian opposite movement scenarios are usually designed to have a relatively open space for pedestrians to move around. However, traditional methods become less capable when pedestrians' action space is compressed. In this study, we design a narrow promenade encounter scenario to investigate the performance of two groups of pedestrians walking towards each other in a narrow area. We randomly place two groups of pedestrians on each side of the promenade, and their respective goals are to cross the promenade and walk to the other side. The length of the promenade is set to 60 and the width to 10, accommodating a maximum of two people moving side by side at the same time. The pedestrian distribution and the expert-guided route are shown in Fig.11(a).
- **Crowded obstacle scenario**. For traditional models, pedestrians are driven to choose the path with

the shortest distance when approaching a target, which tends to result in congestion. Thus, we design a more complex congregation of pedestrians, where four groups of pedestrians are placed in four corners of the environment. Their goals are to reach the other end of their respective diagonal while avoiding obstacles. We want to explore whether DRL can find better action plans and achieve fast convergence with expert guidance. The scene is set as a square with a side length of 70, and the interval between obstacles is set as 10. The pedestrian distribution and the expert-guided route are shown in Fig. 11(b).



(a) Narrow promenade encounter scenario (b) Crowded obstacle scenario

Fig.11 Complex environment

4.2.1 Narrow promenade encounter scenario

The narrow corridor scenario complicates the opposite direction movement scenario in the previous section, as it further reduces the movement space available to the pedestrians and compresses the time for them to adjust their formation. We randomly generate ten pedestrians on each side of the promenade, who need to cross the narrow promenade to reach the other side. Certain behaviors of pedestrians in key frames are presented in Fig.12.

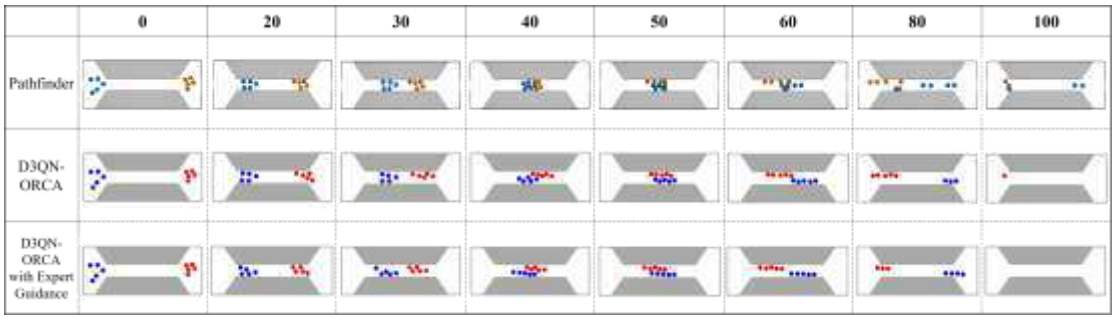


Fig.12 Selected frames in narrow promenade encounter scenario (Pathfinder took a total of 171 frames, D3QN-ORCA took a total of 101 frames, and D3QN-ORCA with expert guidance took a total of 96 frames)

We notice that traditional methods present unsatisfactory performance in this complex scene. Before frame 40, the pedestrians in Pathfinder, driven by the attraction of the target point, always move at the best-desired speed towards the end until two groups of pedestrians meet and a speed barrier region is created. Then, pedestrians start to change their movement direction. However, by this time, the space available for pedestrian movement is limited, and the pedestrians can only fine-tune their respective directions of motion at a slow speed. After 20 frames, crowd congestion remains in the middle of the

promenade. In frame 80, an unreasonable behavior can be observed, where one pedestrian is forced back to the origin. In the end, pedestrians in Pathfinder take 171 frames to complete the simulation.

In contrast, the DRL-based approach takes only about 100 frames to complete the simulation. At frame 20, we notice that pedestrians in D3QN-ORCA, with the addition of expert trajectory guidance, have begun to consciously integrate the pedestrian queue into two columns. At frame 40, the basic D3QN-ORCA presents an orderly queue, with individual pedestrians still needing to avoid minor collisions, while the expert-guided D3QN-ORCA starts to align the crowd, avoiding potential collision future frames. At frame 50, both D3QN-ORCA and D3QN-ORCA with expert trajectory present a smooth pedestrian moving process, where pedestrians are able to maintain their desired speed without considering local dilemmas such as those seen in Pathfinder. Finally, the expert trajectory-guided D3QN-ORCA takes 96 frames for all pedestrians to reach the end point, while the basic D3QN-ORCA does not complete the simulation until 101 frames due to the longer time taken to adjust to the crowd. However, compared to Pathfinder, the DRL-based simulation approach aims to maximize the global reward. While the action taken at each moment may not be optimal at the moment, it should be the optimal global decision.

The reward function curve can also be used to evaluate the effectiveness of the algorithm. To further verify the effectiveness of the expert trajectory guidance, we recorded the average cumulative reward value of all agents at the end of each round, smoothed the curve with a sliding window to enhance the visualization. We compared the change in reward value during training of the basic D3QN-ORCA and expert-guided D3QN-ORCA and explored their convergence speeds (Fig.13).

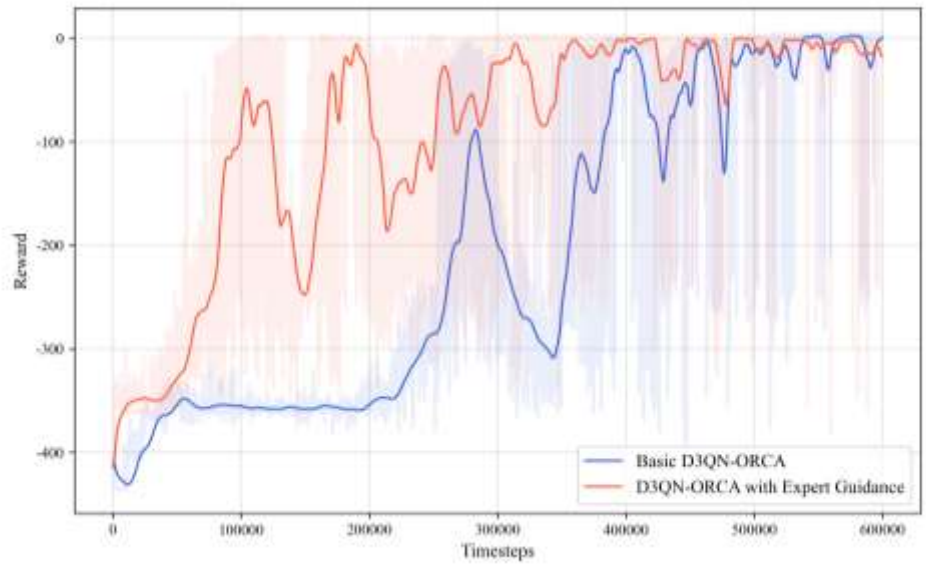


Fig.13 Comparison of D3QN-ORCA reward values with and without expert trajectory guidance.

For a more intuitive comparison, we only introduce the expert trajectory imitation degree reward at training time for the method with expert trajectory guidance, with the sum of the remaining three rewards recorded at plotting time.

From Fig.13, the expert trajectory-guided D3QN-ORCA method gradually explores a better action strategy at around 100,000 steps and converges the reward value steadily around 0 after about 350,000 steps. In contrast, the basic D3QN-ORCA model falls into a local dilemma at the beginning and does not reap a larger reward until around 280,000 steps, and then gradually converges smoothly at around

500,000 steps. Therefore, we can conclude that the expert trajectory guidance can effectively improve exploration efficiency by preventing agents from falling into local dilemmas.

4.2.2 Narrow promenade encounter scenario

In this scene, we placed more groups of pedestrians and obstacles to increase the difficulty of the environment, hoping to explore the behavior patterns of pedestrians in the highly crowded and chaotic scenarios. We generate four pedestrians in each corner of the scene, and their goal is to reach the other side of their respective diagonal. We present simulated screenshots of the current environment every 20 frames to compare model performances (Fig.14).

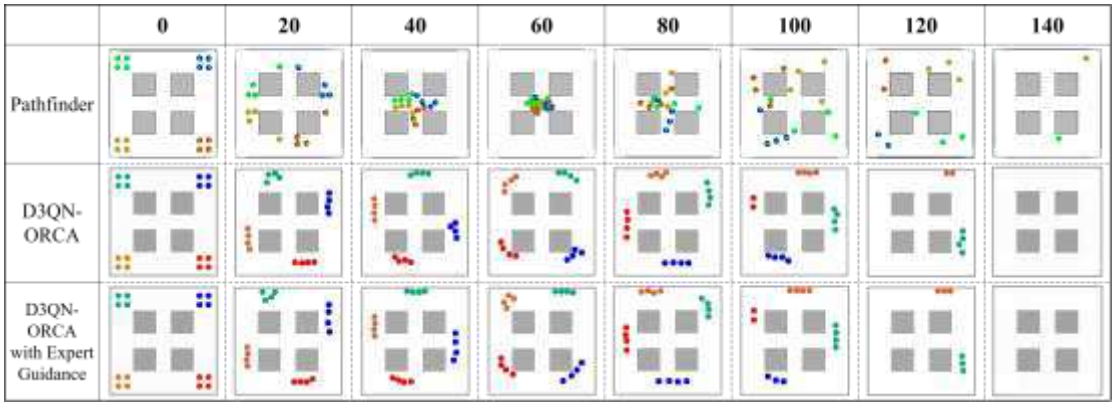


Fig.14 Selected frames in crowded obstacle scenario (the Pathfinder took a total of 157 frames, D3QN-ORCA took a total of 131 frames, and D3QN-ORCA with expert guidance took a total of 130 frames)

It is notable that agents in the traditional method behave differently compared to the ones in DRL-based methods. Given the short-sightedness of the Pathfinder simulation method, pedestrians fail to consider the potential future chaos before a crowded collision and always prefer following the path with the shortest distance. At frame 40, due to the symmetry of the initial positions, four groups of pedestrians arrive at the middle of the scene almost simultaneously, causing congestion of pedestrian flow. The congestion did not ease until after frame 80. It took 157 frames for agents in the Pathfinder model to end the simulation. In comparison, agents in D3QN-ORCA discover the local dilemma after interacting with the environment and keep seeking other optimal solutions. It adopts a movement pattern of four groups of pedestrians going around clockwise synchronously from the beginning. The crowd moves smoothly without congestion until frame 120. As a result, all the pedestrians in the basic D3QN-ORCA reached the target point at frame 131.

Similarly, given the local dilemma that may result from all pedestrians expecting to choose the shortest distance path at the same time, we set the expert-guided trajectory to a simultaneous same-direction detour mode for the four groups of people as well. Pedestrians in D3QN-ORCA with expert-guided trajectory present similar performance compared to basic D3QN-ORCA, with one frame difference when completing the simulation.

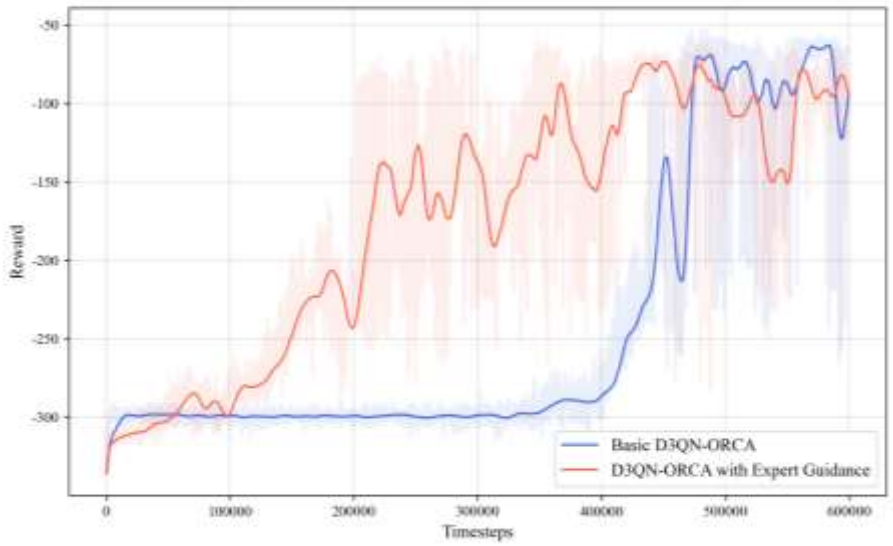


Fig.15 Comparison of D3QN-ORCA reward values with and without expert trajectory guidance.

From Fig.15, we notice that the agents in the D3QN-ORCA method without expert trajectory guidance are to jump out of the local optimum before 400,000 steps, with insufficient exploration. The model starts to gradually converge from 450,000 steps when better behavioral strategies are sampled. On the contrary, in the model guided by expert trajectory, the effective exploration rate of the agents does not have a long stagnant phase, presenting a longer, constantly growing pattern. The global optimal decision action that can be achieved in each state is gradually explored, and the final reward value converges smoothly at around -100.

Table 2 presents the total simulation time consumed by each method in all scenarios. In the simple scenarios, the overall time difference between the two models is about ten frames, while in the complex scenarios, the simulation time based on the DRL method is substantially better than that of Pathfinder. Therefore, it is proved that D3QN-ORCA has great generalizability, and the more complex the environment, the more prominent the advantage of being able to cope with potential congestion situations. Moreover, with the introduction of expert trajectory guidance, the simulation process can be further improved on the basis of enhancing training efficiency. It gives pedestrians certain guidance while encouraging ensuring that they are left with the space for free exploration and the characteristics of reinforcement learning for environmental interaction sampling to achieve optimization of pedestrian simulation.

Table 2 Total simulation time consumed by each method in all scenarios (Unit: Frame)

Model	Simple environment		Complex environment	
	Room evacuation	Opposite motion	Narrow promenade encounter	Crowded obstacle
Pathfinder	138	129	171	157
Basic D3QN-ORCA	**127**	**112**	101	131
D3QN-ORCA with expert guidance	—	—	**96**	**130**

5 Conclusion

Pedestrian simulation has received extensive attention over the past decades. Traditional pedestrian simulation methods often suffer from the problem of short-sightedness, as agents prefer the best action in the present moment without taking into account the potential congestion in the future. In this study, we combine the D3QN algorithm with the ORCA algorithm and propose a D3QN-ORCA hierarchical model. The proposed model plans the global trajectory via the D3QN algorithm in the upper layer while avoiding local collisions among individuals using the ORCA algorithm in the bottom layer. For certain scenarios where local optimum tends to occur, we introduce expert trajectory imitation degree in the reward function, which improves pedestrian exploration efficiency by designing simple expert trajectory guidance lines without collecting priori sample sets and constructing complex expert example databases. We design two different experimental scenarios, i.e., simple scenarios and complex scenarios, to verify the effectiveness of the proposed method. In simple scenarios, by comparing the overall simulation frames and pedestrians' behaviors, we notice that the D3QN-ORCA method proposed in this study is superior compared with the Pathfinder. The agents in the D3QN-ORCA method can make early adjustments for the upcoming congestion, which reduces the overall simulation time by about 8%-13%. In complex scenarios, the results suggest that our method presents great generalizability, and the overall simulation time is reduced by about 17%-44% compared to the Pathfinder. In addition, based on the reward curve graph, we notice that the D3QN-ORCA model with expert trajectory guidance improves the exploration efficiency, evidenced by the fact that the time from the initial exploration to the maximum cumulative reward value is reduced by about 56%-64%. Expert guidance facilities pedestrians in terms of moving out of local dilemmas, thus leading to improved training efficiency. The proposed conceptual, theoretical, and experimental knowledge this study presents is expected to benefit future pedestrian simulation and crowd evacuation studies.

Declarations

Ethics approval and consent to participate: Not applicable

Availability of data and material: All of the material is owned by the authors and/or no permissions are required.

Competing interests: The authors have no competing interests to declare that are relevant to the content of this article.

Funding: Not applicable

Authors' contributions: S.M. and X.L. conceived and designed the experiments. S.M., X.H., M.W., D.Z., D.X. performed the experiments and analyzed the results. S.M., X.H., and M.W. wrote the manuscript. X.L. gave comments and suggestions on the manuscript and proofread the document. All authors have read and agreed to the published version of the manuscript.

Acknowledgements: We thank the editors and reviewers for their valuable comments.

References

[1] Xu D (2021) Study on micro-scale pedestrian simulation using reinforcement learning [D]. East China Normal University. DOI: 10.27149/d.cnki.ghdsu.2021.000427.

[2] Du J (2020) Research on emergency evacuation modeling and path planning based on artificial bee colony algorithm [D]. Hubei University of Technology. DOI: 10.27131/d.cnki.ghugc.2020. 000042.

[3] D Helbing, Farkas I J, Molnar P, et al. (2002) Simulation of pedestrian crowds in normal and evacuation situations[M].

[4] Dijkstra J, Jessurun J, Timmermans H J P (2001) A multi-agent cellular automata model of pedestrian movement[J]. Pedestrian and evacuation dynamics, 173: 173-180.

[5] Helbing D (1998) A fluid dynamic model for the movement of pedestrians[J]. arXiv preprint cond-mat/9805213.

[6] Wu Z, Liu D, Cheng Y, Sun Y (2012) Three-dimensional crowd simulation of agent-based method[J]. Computer technology and development, 22(11): 108-112.

[7] Zhao L, Guo M, Tang S, Tang J (2022) Adaptive crowd evacuation simulation model based on bounded rationality constraints [J/OL]. Journal of system simulation: 1-9. DOI: 10.16182/j. issn1004731x. joss. 21-0472.

[8] Shen Y, Han J, Li L, et al (2020) AI in game intelligence—from multi-role game to parallel game[J]. Chinese Journal of Intelligent Science and Technology, 2(3): 205-213.

[9] Hu Y, Mottaghi R, Kolve E, et al (2017) Target-driven visual navigation in indoor scenes using deep reinforcement learning[C]//2017 IEEE international conference on robotics and automation (ICRA). Piscataway: IEEE Press, pp 3357-3364.

[10] Shani G, Heckerman D, Brafman R I, et al (2005) An MDP-based recommender system[J]. Journal of Machine Learning Research, pp 1265-1295.

[11] Yao Z (2020) Research on simulation method of crowd evacuation based on reinforcement learning and deep residual network learning [D]. Shandong Normal University. DOI: 10.27280/d.cnki. gsdsu.2020.001531.

[12] Xu D, Huang X, Li Z, et al (2020) Local motion simulation using deep reinforcement learning[J]. Transactions in GIS, 24(3): 756-779.

[13] Lowe R, Wu Y I, Tamar A, et al (2017) Multi-agent actor-critic for mixed cooperative-competitive environments[J]. Advances in neural information processing systems, 30.

[14] Zhang F, Li J, Li Z. A (2020) TD3-based multi-agent deep reinforcement learning method in mixed cooperation-competition environment[J]. Neurocomputing, 411: 206-215.

[15] Zhelo O, Zhang J, Tai L, et al (2018) Curiosity-driven exploration for mapless navigation with deep reinforcement learning[J], arXiv: 1804.00456.

[16] Yang Y (2019) Study on Crowd Evacuation Simulation Models for Semi-Submersible Accommodation Platform [D]. Shanghai Jiao Tong University. DOI: 10. 27307/d.cnki.gsjtu.2019. 001894.

[17] Bounini F, Gingras D, Pollart H, et al (2017) Modified artificial potential field method for online path planning applications[C]//2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp 180-185.

[18] Wu H (2017) Evacuation Simulation of Indoor Pedestrian [D]. University of Electronic Science and Technology of China.

[19] Ma S, Zhang R, Qi Z, Hao J (2021) Research on improvement of social force model of opposite avoidance and contact behavior [J]. Computer simulation, 38(03): 63-67+77.

[20] Guo K, Wang D, Fan T, et al (2021) VR-ORCA: Variable Responsibility Optimal Reciprocal Collision Avoidance[J]. IEEE Robotics and Automation Letters, 6(3): 4520-4527.

[21] He G, Jiang D, Jin Y, Chen Q, Lu X, Xu M (2018) Crowd behavior simulation based on shadow obstacle and ORCA models [J]. SCIENTIA SINICA Informationis, 48(03):233-247.

[22] Silver D, Huang A, Maddison C J, et al (2016) Mastering the game of Go with deep neural networks and tree search[J]. nature, 529(7587): 484-489.

[23] Mnih V, Kavukcuoglu K, Silver D, et al (2015) Human-level control through deep reinforcement learning[J]. nature, 518(7540): 529-533.

[24] Van Hasselt H, Guez A, Silver D (2016) Deep reinforcement learning with double q-learning[C]//Proceedings of the AAAI conference on artificial intelligence. 30(1).

[25] Wang Z, Schaul T, Hessel M, et al (2016) Dueling network architectures for deep reinforcement learning[C]//International conference on machine learning. PMLR, pp 1995-2003.

[26] Schaul T, Quan J, Antonoglou I, et al (2015) Prioritized experience replay[J]. arXiv preprint arXiv:1511.05952.

[27] Williams R J (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine learning, 8(3): 229-256.

[28] Schulman J, Levine S, Abbeel P, et al (2015) Trust region policy optimization[C]//International conference on machine learning. PMLR, pp 1889-1897.

[29] Schulman J, Wolski F, Dhariwal P, et al (2017) Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347.

[30] Lillicrap T P, Hunt J J, Pritzel A, et al (2015) Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971.

[31] Fujimoto S, Hoof H, Meger D (2018) Addressing function approximation error in actor-critic methods[C]//International conference on machine learning. PMLR, pp 1587-1596.

[32] Haarnoja T, Zhou A, Abbeel P, et al (2018) Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//International conference on machine learning. PMLR, pp 1861-1870.

[33] Lee J, Won J, Lee J (2018) Crowd simulation by deep reinforcement learning[C]//Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games. pp 1-7.

[34] Sharma J, Andersen P A, Granmo O C, et al (2020) Deep q-learning with q-matrix transfer learning for novel fire evacuation environment[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(12): 7363-7381.

[35] Lu G (2021) Regularized maximum entropy imitation learning based on prior reward of trajectory [D]. East China Normal University. DOI: 10.27149/d.cnki.ghdsu.2021.000029.

[36] Levine S, Kumar A, Tucker G, et al (2020) Offline reinforcement learning: Tutorial, review, and perspectives on open problems[J]. arXiv preprint arXiv:2005.01643.

[37] Berg J, Guy S J, Lin M, et al (2011) Reciprocal n-body collision avoidance[M]//Robotics research. Springer, Berlin, Heidelberg, pp 3-19.

[38] Fortunato M, Azar M G, Piot B, et al (2017) Noisy networks for exploration[J]. arXiv preprint arXiv:1706.10295.