中国科学院材料力学行为和设计重点实验室

CAS Key Laboratory of Mechanical Behavior and Design of Materials

An efficiently convergent deep reinforcement learning-based trajectory planning method for manipulators in dynamic environments

Li Zheng, YaHao Wang, Run Yang, Shaolei Wu, Rui Guo and Erbao Dong

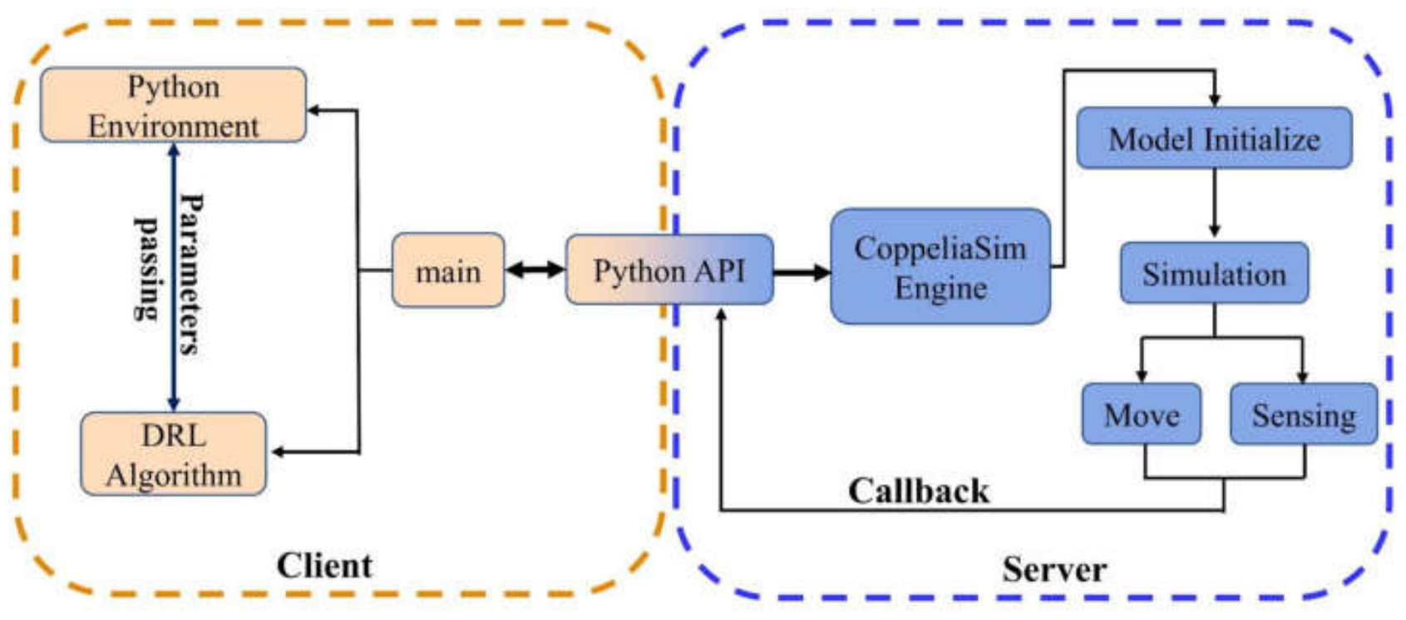
University of Science and Technology of China

INTRODUCTION

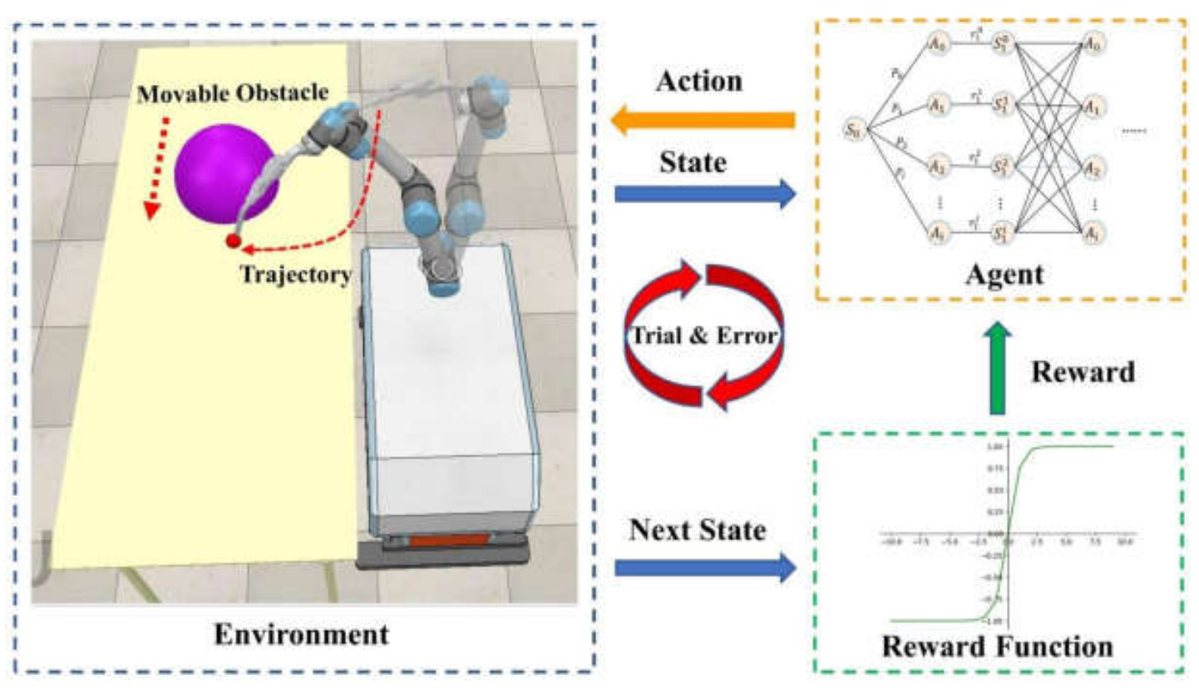
Recently, deep reinforcement learning-based trajectory planning methods have been designed for manipulator trajectory planning, given their potential in solving the problem of multidimensional spatial trajectory planning. However, many DRL models that have been proposed for manipulators working in dynamic environments face difficulties in obtaining the optimal strategy, thereby preventing them from reaching convergence because of massive ineffective exploration and sparse rewards. This study solved the inefficient convergence problem at the two levels of the action selection strategy and reward functions. First, a dynamic action selection strategy was designed to enhance the likelihood of yielding favorable samples during the pre-training phase. This was achieved through the incorporation of a variable guide item, effectively mitigating instances of invalid exploration. Second, a composite reward function was introduced, integrating the artificial potential field method with a time-energy function. This integration markedly enhances the efficiency and stability of DRL-based techniques for manipulator trajectory planning in dynamic operational contexts.

PRINCIPLE AND DESIGN

1. This study designed a co-simulation framework of CoppeliaSim and Python *(a)*. The co-simulation framework uses Python API to interact with Python clients through CoppeliaSim's remote API client mode. Through the remote API client mode, different DRL algorithms can be combined to train the simulated robot model *(b)*.

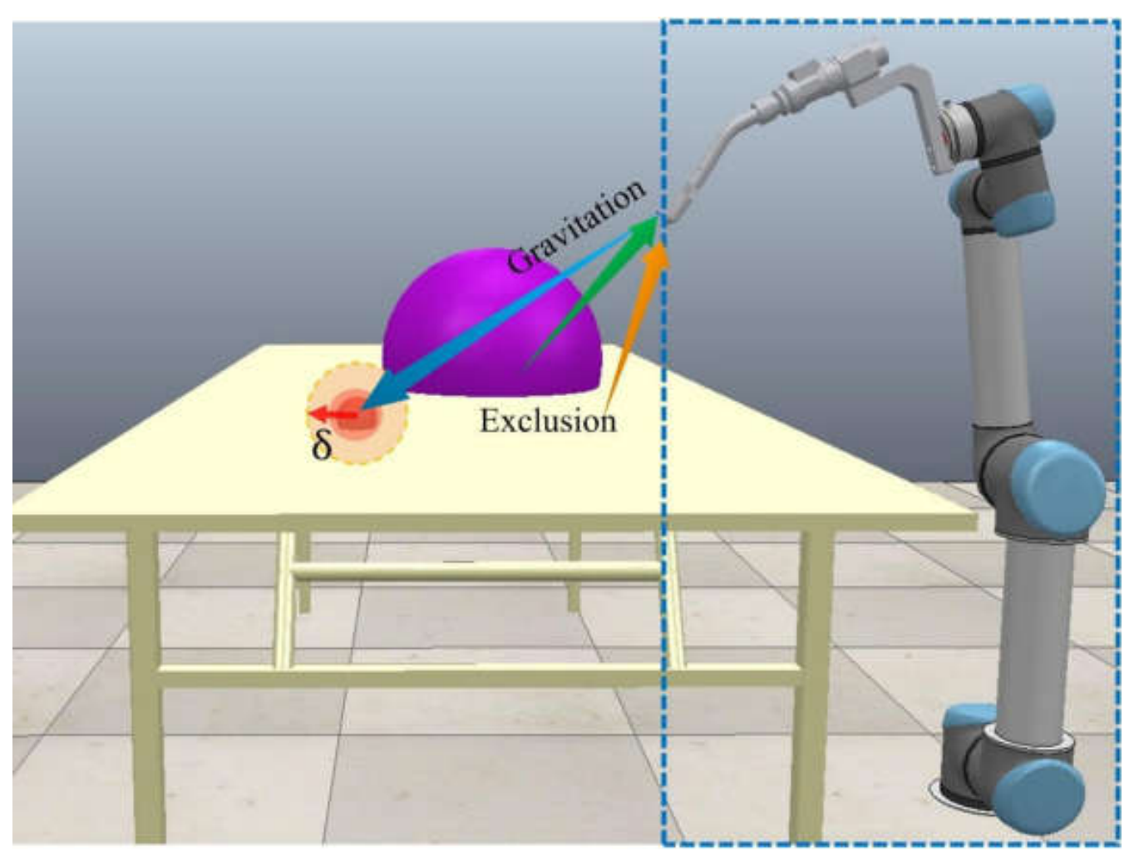


(a)

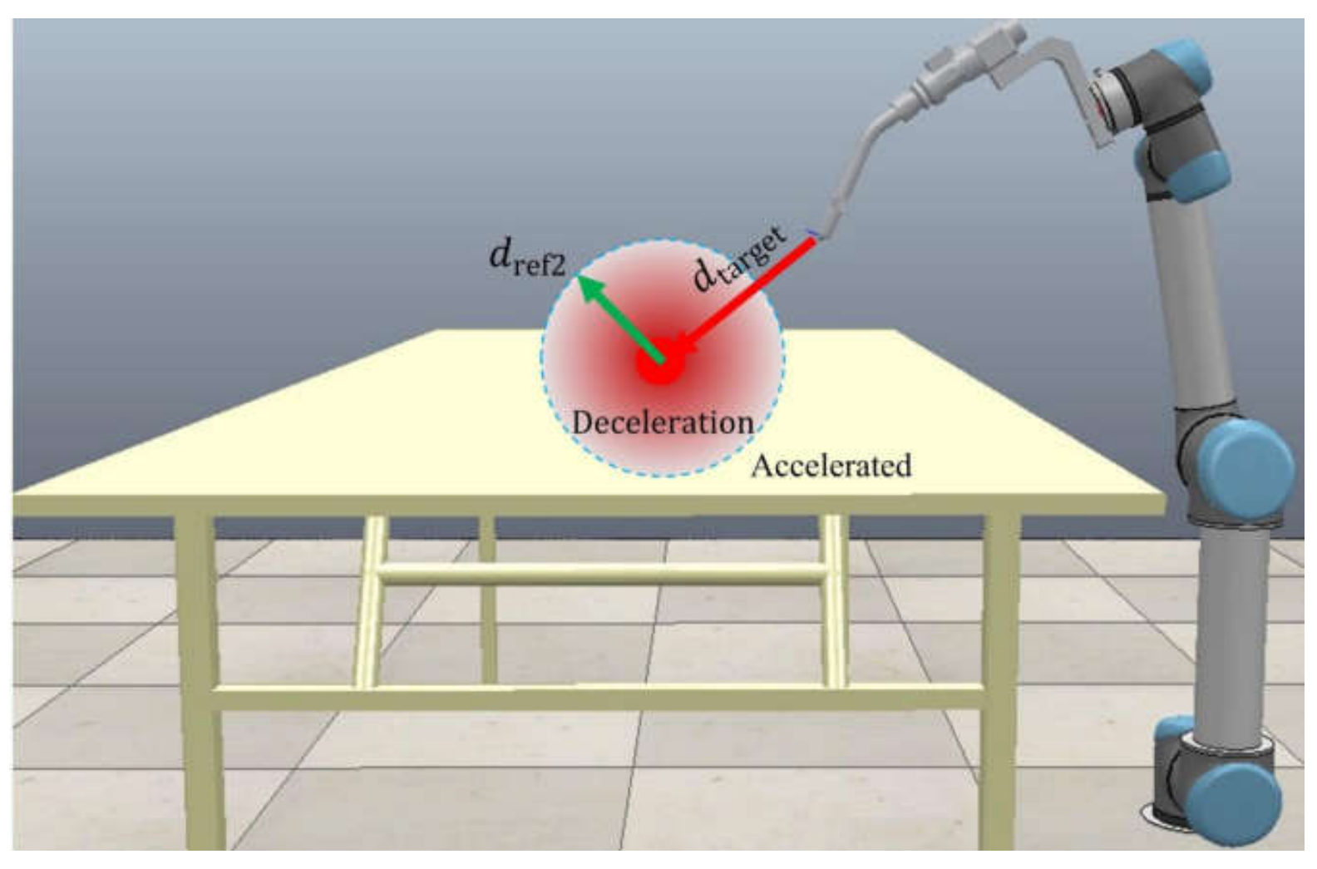


(b)

2. To solve the sparse reward problem of DRL-based methods for manipulator trajectory planning in dynamic environments, a combinatorial reward function that incorporates the ideas of the artificial potential field method *(c)* and the time-energy function *(d)* was proposed. The reward function is a scalar function comprising a four-term sum and is defined as follows: the target attraction reward function R_{target}, the obstacle rejection reward function $R_{obstacle}$, the energy reward function R_{energy}, and the time reward function R_{time}.



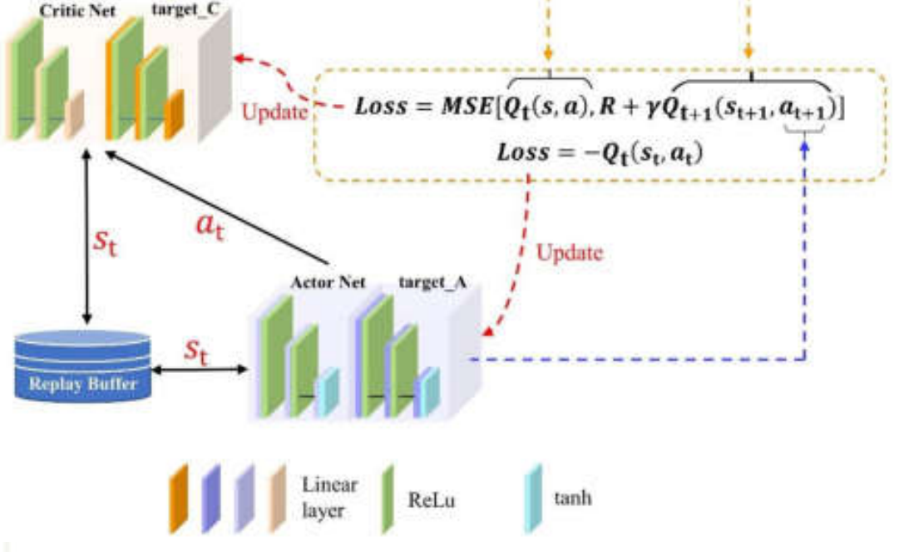
(c)



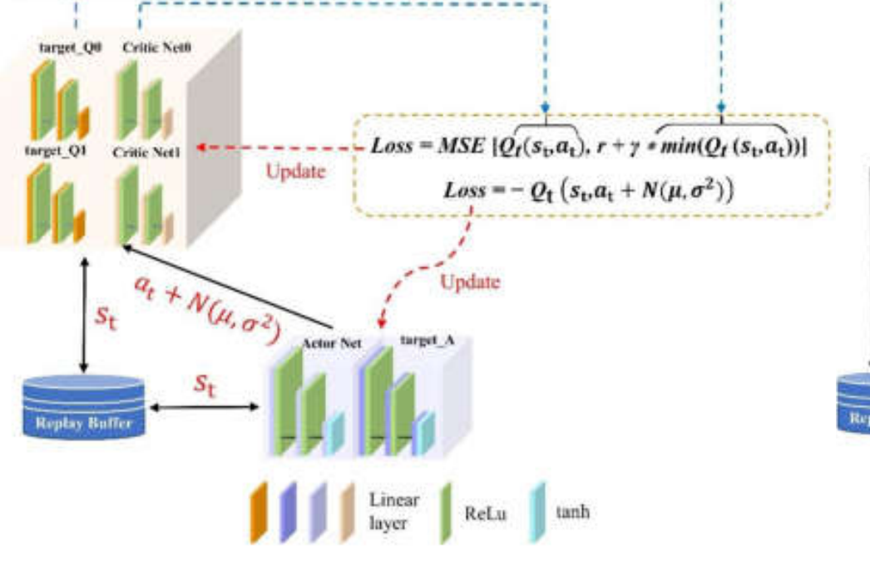
(d)

METHOD

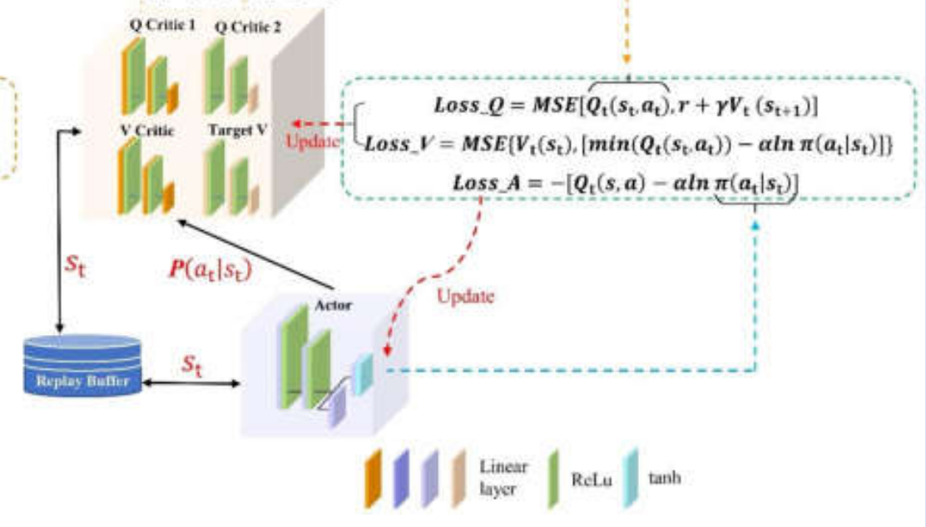
1. In this study, the policy learning algorithms of the agent are DDPG, TD3, and SAC *(e-g)*.



(e)

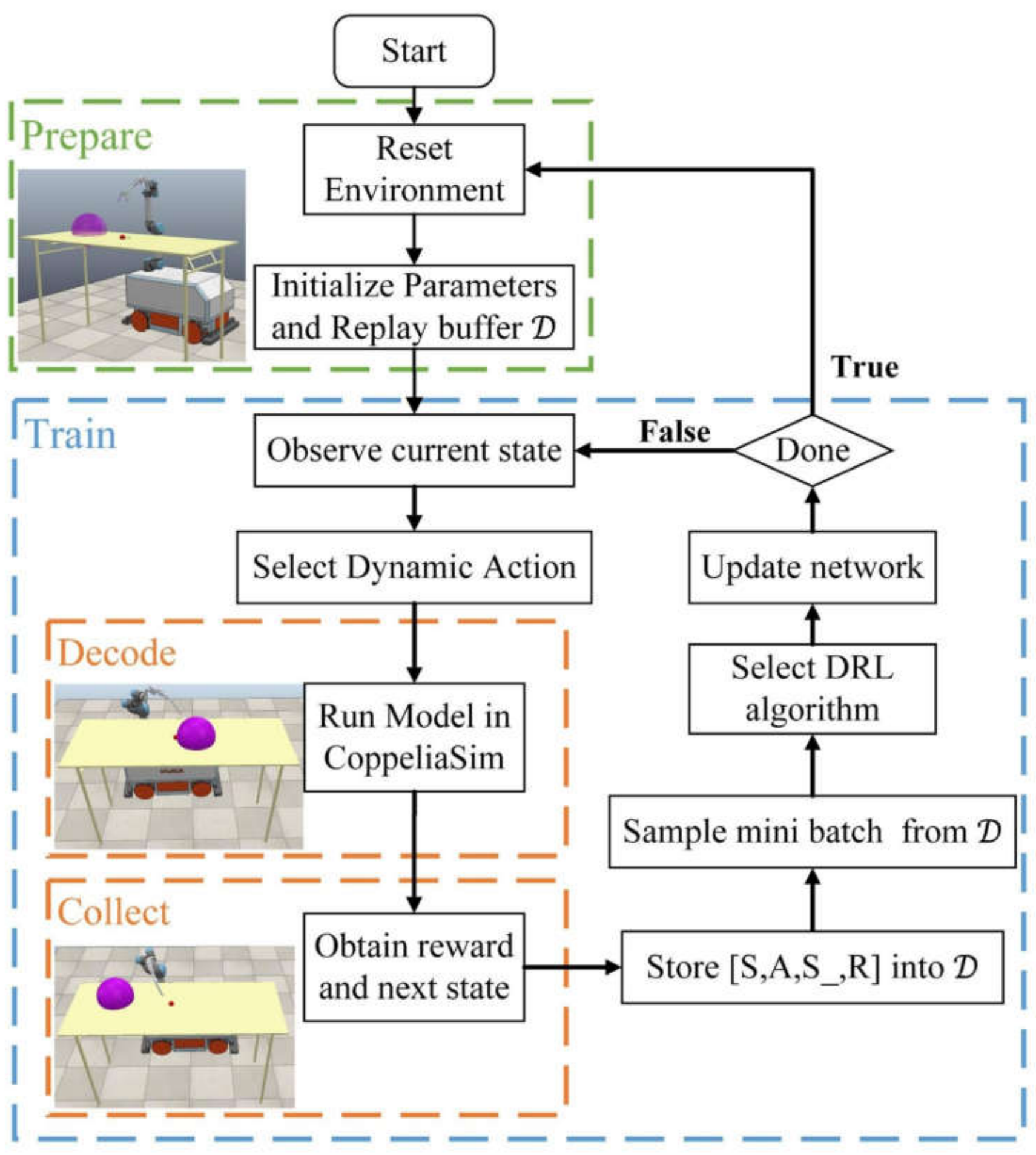

$min(Q_t^{\theta 0}(s,a), Q_{t+1}^{\theta 1}(s,a))$
target_Q0
Critic Net0
target_Q1
Critic Net1
Update
$Loss = MSE[Q_t(s_t,a_t), r + \gamma * min(Q_t(s_t,a_t))]$
$Loss = -Q_t(s_t, a_t + N(\mu, \sigma^2))$
$a_t + N(\mu, \sigma^2)$
s_t
Actor Net
target_A
Replay Buffer
Linear layer
ReLu
tanh

(f)


$Q_t^{\theta 1}(s,a)$
$Q_t^{\theta 2}(s,a)$
Q Critic 1
Q Critic 2
V Critic
Target V
Update
$Loss\_Q = MSE[Q_t(s_t,a_t), r + \gamma V_t(s_{t+1})]$
$Loss\_V = MSE\{V_t(s_t), [min(Q_t(s_t,a_t)) - \alpha \ln \pi(a_t|s_t)]\}$
$Loss\_A = -[Q_t(s,a) - \alpha \ln \pi(a_t|s_t)]$
s_t
$P(a_t|s_t)$
Actor
Replay Buffer
Linear layer
ReLu
tanh

(g)

2. All simulation training processes *(h-g)* for the entire model were run on a computer with an 8-core Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz, 16.0 GB RAM, and an NVIDIA GeForce RTX 3060 GPU.



(h)

Algorithm 1 Trajectory Planning Algorithm with Dynamic Action and Reward Function

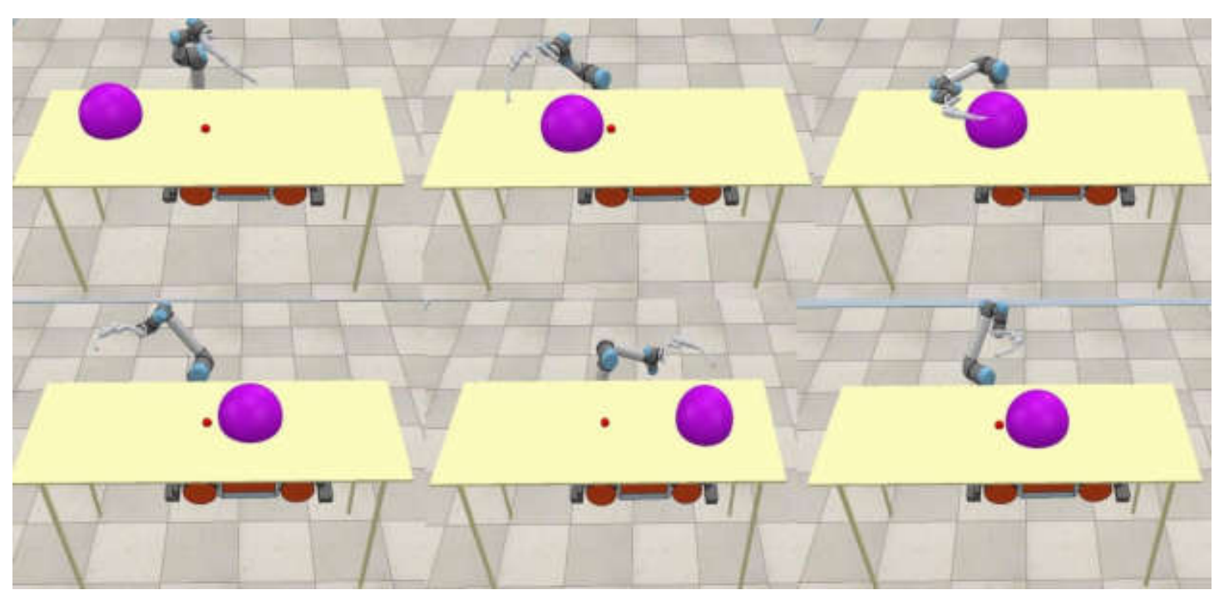
```
Require: Environmental status S;
Ensure: Action
1: Initialize Networks parameters and Replay buffer $\mathcal{D}$;
2: for current episode = 1 to MAX EPISODE do
3:     for current step = 1 to MAX STEP do
4:         $a_t \leftarrow a_t \cdot rate + \Delta q \cdot (1 - rate)$
5:         $s_{t+1} \leftarrow p(s_{t+1} \mid s_t, a_t)$
6:         Compute $R_{\text{target}}, R_{\text{obstacle}}, R_{\text{energy}}, R_{\text{time}}$
7:         $R = R_{\text{target}} + R_{\text{obstacle}} + R_{\text{energy}} + R_{\text{time}}$
8:         $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, R, s_{t+1})\}$
9:         if $\mathcal{D}$ size > batch size then
10:            Select DRL Algorithm
11:            Update weigh of Networks
12:        end if
13:    end for
14: end for
```

(i)

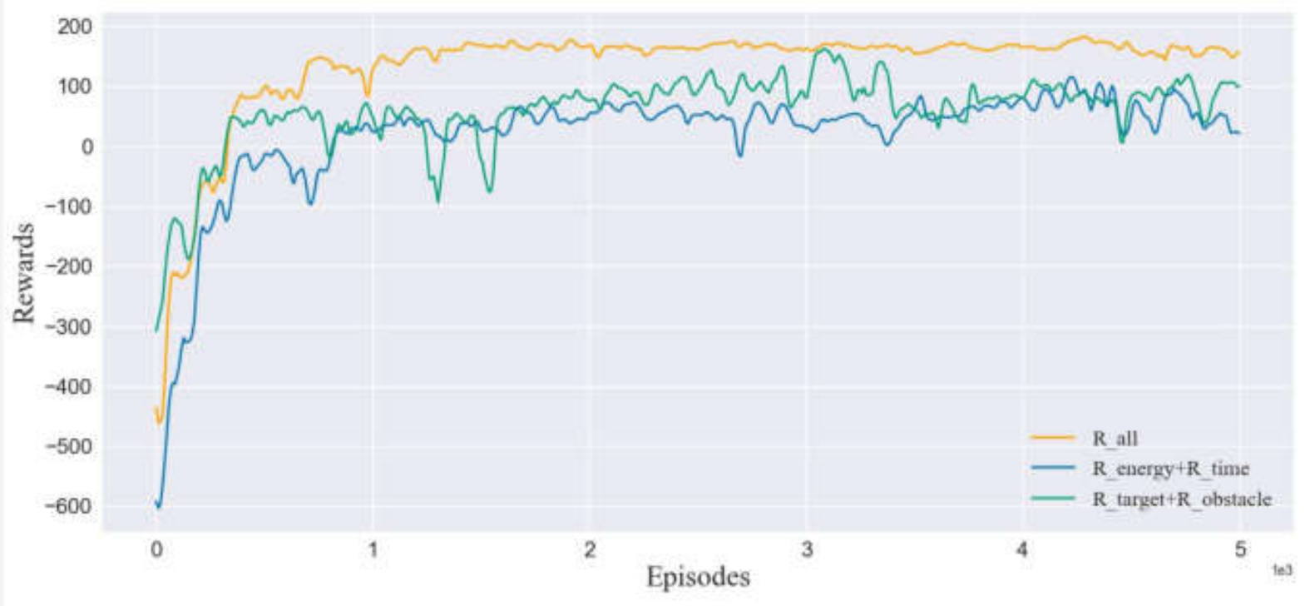
EXPERIMENTS AND RESULTS

1. Six groups of experiments (j-m) were conducted to test the performance of the proposed dynamic action selection strategy and combinatorial reward function. The performance of these methods was evaluated using three indicators, namely, convergence rate, success rate, and mean reward value.

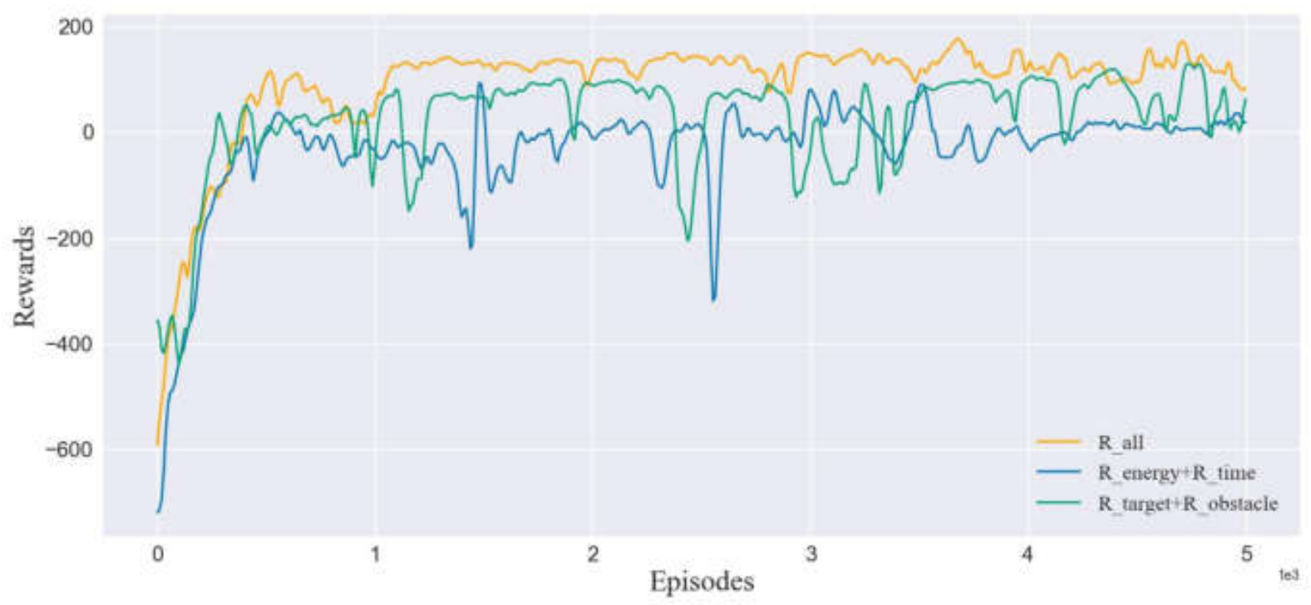
(j)



(k)

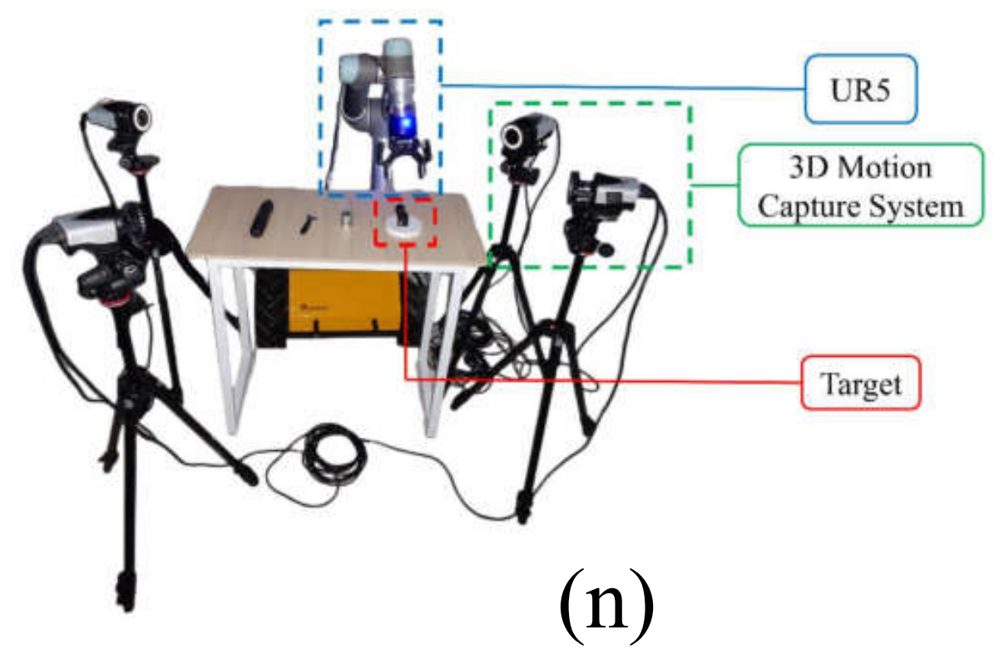


(l)

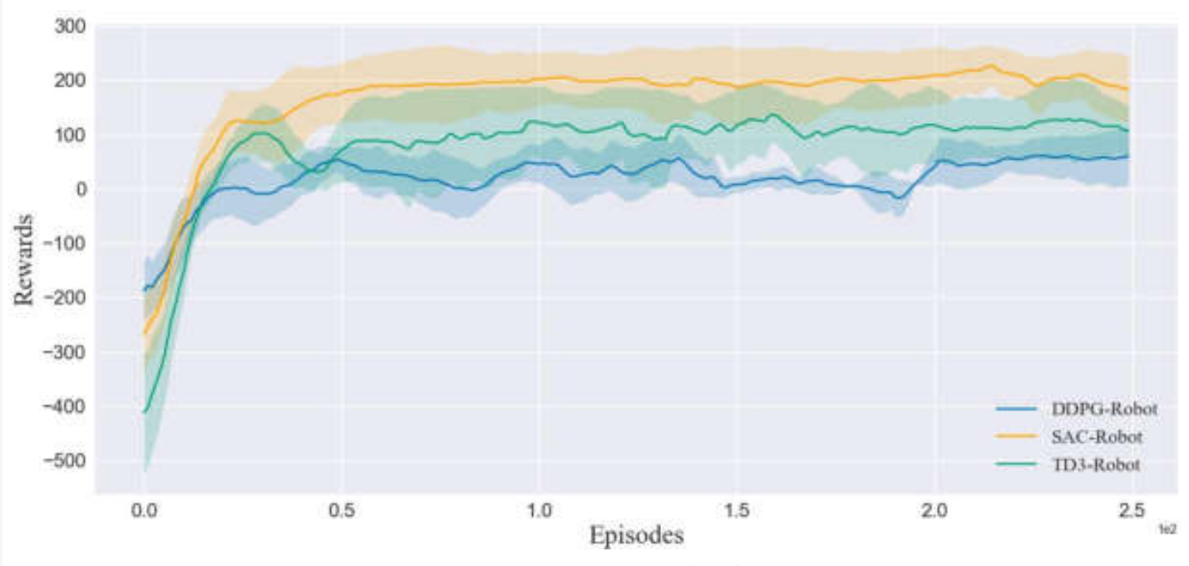


(m)

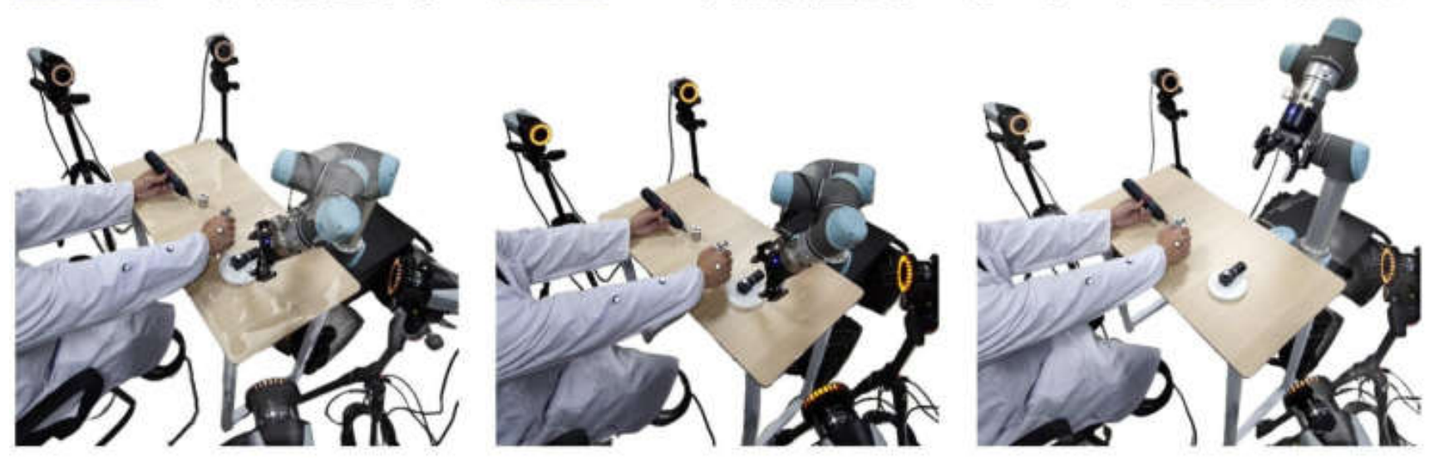
2. Guided by the selection of hyper-parameters for the proposed algorithm and the networks designed in the simulation environment, the proposed dynamic action selection strategy and the combined reward function are evaluated and verified in a realistic dynamic obstacle environment (n-p).



(n)



(o)



(p)

CONCLUSIONS

This study proposed an efficiently convergent DRL-based trajectory planning method for manipulators in dynamic environments. This method can be applied to autonomous navigation and real-time obstacle avoidance tasks, avoiding dynamic modeling and parameter optimization processes. Multiple experiments on three algorithms were performed for comparison, and the results show that the proposed dynamic action selection strategy and combinatorial reward function can greatly improve the convergence rate by 3-5 times and increase the success rate by 72.72%-82.62%.

USTC-LMBD