Prophet Actor-Critic-Based Deep Reinforcement Learning for Obstacle Avoidance in Robotic Arm Control

Jiangtao Wang
College of Electrical and Control Engineering, Nanjing Polytechnic Institute, Nanjing 210048, China
E-mail: wangjiangtao@njpi.edu.cn

Keywords: deep reinforcement learning, robotic arm, intelligent model, AI control

Received: 2025.1.9

For the current common obstacle avoidance trajectory planning tasks of robotic arms, the problem of insufficient versatility is usually encountered. The grasping performance of the manipulator is mainly constrained by obstacles. How to improve the obstacle avoidance ability of the manipulator to improve its grasping ability. In order to improve the intelligent control effect of the robotic arm, an intelligent AI control method of the robotic arm combined with deep reinforcement learning is proposed. Moreover, in order to solve the problems of low learning efficiency caused by low-quality empirical data in the early stage of training and low efficiency in obtaining expert empirical data, an EDS mechanism is proposed to improve training efficiency by expanding expert empirical data and adopting unbiased dual memory bank sampling rules. In addition, in order to enhance the obstacle avoidance capability of the robotic arm, a COR system is constructed to help quickly generate the optimal trajectory, and the end effector and fuselage of the robotic arm can simultaneously avoid obstacles in complex environments, and achieve a balance between obstacle avoidance and motion exploration. The results show that the success rate of grasping complex objects in both obstacle free and obstacle free environments can reach more than 80%. Compared with the existing models, it has faster convergence speed and learning effect, and has better model performance. This paper combines experimental analysis to verify the effectiveness of the AI control method proposed in this paper, which can effectively ensure the collision-free trajectory planning of the robotic arm in different scenarios and has strong adaptability to scene changes.

Povzetek: Raziskano je globoko ojačevalno učenje za načrtovanje poti in izogibanje ovir pri robotiziranem ročnem nadzoru, kjer kombinacija algoritmov za izboljšano učenje in optimizacijo omogoča hitro prilagajanje v dinamičnih okoljih ter izboljša sposobnost robota za izogibanje oviram in natančno izvajanje nalog.

1 Introduction

With the continuous development of deep learning technology and computer hardware foundation, artificial intelligence algorithms represented by computer vision have made unprecedented breakthroughs. Image processing algorithms based on convolutional neural networks far exceed traditional algorithms in real-time image classification, target detection, semantic segmentation and other scene detection effects, providing new possibilities for the development of related industries. Therefore, adding a visual servo control system to the robotic arm, collecting images through visual sensors, and performing grasping detection based on convolutional neural network will improve the intelligent perception ability of the robotic arm in the working environment [1].

Robotic arm grasping is a complex task involving object detection, robotic arm path planning and target grasping pose estimation. At present, many methods focus on target detection and grasping pose estimation. Among them, a common method is to detect the geometric features of the object to find the suitable grasping position, such as handle-like objects and symmetrical objects, and then estimate the grasping pose of the object, and control the end effector to complete the target grasping. However, many objects do not have similar distinct features, or the target objects may be in a cluttered stacked environment, and the target objects are partially blocked by obstacles, so the vision system cannot accurately detect the target and locate the parts to be grasped. Therefore, in recent years, scholars usually use machine learning methods such as convolutional neural networks [2], which can more robustly calculate the features of targets, so as to realize target detection and target grasping pose.

With the upsurge of artificial intelligence technology research, artificial intelligence technology represented by deep learning and deep reinforcement learning has been widely used in the field of robot research, which has brought new opportunities for the development of robots [3]. In the robot arm collision detection, the sensor method first introduces the deep

learning technology, uses the neural network to learn the signal data of the sensor, and uses the excellent feature extraction ability of deep learning to extract the collision features in the signal, which finds a new solution to the problems of high system complexity and complex detection model establishment in the sensor method [4]. At present, the geometric simulation method has not been developed with depth.

Through the analysis of the research status of capture detection, it can be found that because the capture detection task is more complex than the target detection, and the capture scene is quite different, the production cost of data set is high, so the capture detection neural network based on the capture data set training is relatively difficult and costly in the actual training [5]. Deep reinforcement learning has made some progress in solving the problem of robot arm control and obstacle avoidance. However, there are still some problems such as poor training efficiency, large error and poor adaptability to environmental changes [6]. In addition, due to the long training time of the deep reinforcement learning algorithm, the physical training method has low efficiency and poor security [7]. However, the simulation training method has the problem of poor physical transfer effect, which has brought great trouble to the practical application of deep reinforcement learning algorithm. Therefore, the subsequent capture detection module in this paper will estimate the 3D capture pose of the object based on the feature extraction technology of deep reinforcement learning.

Aiming at common robotic arm grasping scenarios, this paper proposes a robotic arm grasping method based on deep reinforcement learning to solve the robotic arm grasping tasks in corresponding scenarios. Aiming at the multi-target disordered grasping task in cluttered environment, it uses the disordered grasping method in continuous action space. Aiming at the obstacle avoidance capture task in the cluttered stacking environment, it uses the obstacle avoidance capture method in the discrete action space. Moreover, this paper combines two methods to construct a robotic arm grasping system to solve the hybrid grasping task under the above-mentioned environment integration, which has great potential in the field of scientific research and social application value.

With the rapid development of industrial automation and intelligent robot technology, the manipulator system has been widely used in storage and logistics, precision assembly, medical surgery and other fields. However, in the real operation scene, the manipulator often faces complex environmental interference such as dynamic obstacles, illumination changes, target occlusion, which leads to significant limitations of the traditional motion planning algorithm. Statistics show that in the industrial scene with more than 30% random obstacles, the success rate of the 6-DOF manipulator is generally lower than 65%, and the collision risk rate is as high as 22%. There are three main technical bottlenecks: first, the traditional path planning algorithm based on geometric model is difficult to deal with the real-time updated environmental information of dynamic obstacles; Secondly, the conventional vision algorithm is susceptible to texture interference and the lack of depth information in the scene of densely stacked objects, resulting in the error of target pose estimation exceeding ± 5 °; Third, the existing obstacle avoidance strategies mostly use static threshold judgment, and lack of collaborative optimization of manipulator kinematics constraints and end effector grasping posture.

This study proposes a collaborative optimization framework based on deep reinforcement learning, which aims to break through the problem of precise grasping of manipulator in complex environment.

The purpose of this paper is to improve the robot's obstacle avoidance effect based on the existing model, and to solve the problem of customer service data training, so as to improve the robot's intelligent control effect and autonomous decision-making ability, as follows:

(1) In order to solve the problem of poor universality of expert strategy, this paper studies and designs a prophet Strategy Network Guided deep reinforcement learning algorithm. The algorithm has the ability of expert strategy self optimization, can avoid the situation that the expert strategy itself is a local optimal strategy, and can be widely used in changing scenarios. In addition, the guidance effect of expert strategy on different task scenarios is studied and analyzed.

(2) To solve the problem of low efficiency in the use of expert experience data, this paper proposes an unbiased dual memory sampling mechanism combined with the expert memory amplification mechanism. In addition, the guiding effect of the mechanism in the early stage of algorithm training and the influence of algorithm convergence are studied.

2 Related works

In order to better apply deep reinforcement learning algorithms to real robotic arms, Mourtzis et al. [8] proposed a method that combines convolutional neural network with deep reinforcement learning to perform robot control tasks that require close correlation between vision and control. It uses a monocular camera for image acquisition as the original input, performs feature extraction through a convolutional neural network, and then directly inputs the feature information into deep reinforcement learning to achieve end-to-end joint training of perception and control systems. Moreover, it trains the model in a simulation environment and then migrates to the real manipulator control task. The experimental results show that this end-to-end method can perform complex operation skills, but it is very unstable and takes a lot of time to train in the network.

Zhou et al. [9] proposed a Robotics Transformer (RT-1) algorithm framework, which can effectively absorb a large amount of data and expand with the amount and diversity of data. By using 13 robotic arms to collect more than 130k sets of large demonstration data sets in 17 months to train RT-1, the algorithm has achieved very good results. Through many experiments, the new algorithm has achieved a success rate of 97% in various control tasks of robotic arms, such as grasping and handling, and the new algorithm can be effectively generalized to new tasks, objects and environments.

Singh et al. [10] proposed a learning-based hand-eye coordination robotic arm grasping method. The method is data-driven, and it centers on the object to be grasped, directly from the image pixel to the robotic arm end effector motion, and performs end-to-end network training. In order to better apply the algorithm to real scenarios, the data demand of end-to-end network training is explored and tested through experiments, and the effectiveness of this method is verified. Then, the network model is trained by using 12%, 25% and 50% of the captured data in the dataset. The results show that with the increasing number of captured data, the capturing success rate of the trained network model continues to improve. Although such large-scale and long-term training has great guiding significance in exploring the potential of the algorithm, it also proves that it takes a lot of time and cost to train the model of the end-to-end framework.

In order to better solve the problem of difficult training of vision-based deep reinforcement learning, Abdullah-Al-Noman et al. [11] proposed a modular deep reinforcement learning method, which can transfer the simulation trained model to real-world robot tasks. By introducing a network connection layer between the two modules, the visual inspection network and the control network of deep reinforcement learning are modified to enable the network to be trained independently. When the independent modules are trained, they are merged and fine-tuned in an end-to-end manner to further improve hand-eye coordination.

The model-free algorithm based on deep reinforcement learning shows strong autonomous decision-making ability without the need for a system model, which provides new ideas for robotic arm path planning and target grasping pose estimation. Xu et al. [12] proposed a method of global path planning and local reinforcement learning to avoid obstacles. In the motion of global path planning, reinforcement learning is used to avoid obstacles according to local environmental information. In addition, deep Q-value learning and dual deep Q-value learning algorithms also realize the path planning of the robotic arm [13], process the spatiotemporal information of the system, and solve the path planning and obstacle avoidance problems of the robot in the dynamic environment. Tang et al. [14] used deep Q value to learn the grasping pose estimation of the object, and only when the jaw successfully grasps the object, the agent is given a positive reward value. Nowadays, more advanced discrete reinforcement learning and continuous reinforcement learning algorithms [15] have become the mainstream of training robotic arm control strategies in deep reinforcement learning algorithms, and both have realized a series of robotic arm grasping tasks. However, due to problems such as low sample training efficiency [16], deep reinforcement learning has always had a bottleneck in its wide application in real-world robots. Although there is already an effective position-controlled robotic arm operation framework [17], they still need a lot of time to train each task and provide intensive reward information. Moreover, in the robotic arm grasping task, besides path planning, we should also pay attention to target recognition and grasping pose estimation. Therefore, it is often necessary to perform network training using image input [18]. At the same time, the high-dimensional and complex and changeable image input makes the problems of low efficiency of deep reinforcement learning training samples and difficult network training.

In order to solve the above-mentioned problems of deep reinforcement learning in robotic arm grasping applications, aiming at the problem of image input, Contrastive Unsupervised Representations for Reinforcement Learning (CURL) uses contrastive learning to extract image input and reduce its dimensionality into vector information for the training of evaluation networks and policy networks, which effectively reduces the complexity of training [19]. In order to improve the sample efficiency of deep reinforcement learning and make the learned control strategy more robust, image enhancement is usually used nowadays. Image enhancement is an image-based data enhancement that includes random transformations such as cropping, rotation, or color dithering. It is widely used in computer vision architecture, including pioneering works such as LeNet and AlexNet [20]. However, only recent studies have fully demonstrated the effectiveness of data reinforcement for deep reinforcement learning. Raj et al. [21] proposed a robotic arm control framework combining contrastive learning and image enhancement, which can quickly solve a series of simple robotic arm control tasks, such as grasping squares and opening drawers without barriers. However, it is also difficult to solve complex tasks such as obstacle avoidance and grasping of robotic arms. In addition, the Sim2Real method can also solve the sample efficiency problem in the robotic arm grasping application based on deep reinforcement learning. First, the agent is trained in the simulation environment, and then migrated to the real world for use [22]. In simulation environment training, the physical attributes of the environment can be random, so the images collected by the body vision system are richer, which can effectively increase the training sample size. In addition, imitation learning also plays a role in helping robotic arms train reinforcement learning control

strategies [23]. One of the simplest forms of imitation learning is behavior cloning (BC), which uses regression to fit the expert sample set, so that reinforcement learning agents can learn strategies close to expert control. However, imitation learning usually needs to collect a large number of demonstration samples, and if only BC is used to train the agent, the expert strategy will limit the learning ability of the agent [24], and the best strategy performance of the agent will not be able to surpass the expert strategy.

Table 1: advantages and disadvantages of existing model algorithms

	Algorithm model	Advantage	Deficiency
1	(RT-1) algorithm	High accuracy and strong scalability	Requires a lot of data training, and the effect of path planning is poor
2	Reinforcement learning	Good path planning effect and accurate target capture	The efficiency of training samples is low,
3	Unsupervised reinforcement learning	Good path planning effect, accurate target capture and high training efficiency	Poor obstacle avoidance effect

The advantages and disadvantages of the existing research are shown in Table 1. From Table 1, unsupervised reinforcement learning has the problems that customer service needs a lot of training data and sample training efficiency is low, but the effect of obstacle avoidance is not good, and it still needs to be further improved. Therefore, this paper proposes an improved deep reinforcement learning model to further improve the effect of intelligent obstacle avoidance of manipulator.

3 Robotic arm obstacle avoidance trajectory planning framework based on deep reinforcement learning

The motion path of the manipulator in complex environment will be affected, resulting in its inability to accurately grasp the target. Improving the success rate of the manipulator in cluttered environment through the visual obstacle avoidance algorithm is the research direction of this paper. Through the deep reinforcement learning to improve the visual obstacle avoidance effect, provide reliable motion path and grasp direction for the manipulator, and improve the success rate of the manipulator.

In this chapter, a general trajectory planning framework based on DRL is proposed to realize the autonomous obstacle avoidance of robot tasks. A prophet guided actor critic structure based on expert strategy is designed to support rapid re planning of work scene changes. Secondly, an extended double memory sampling mechanism is proposed to effectively expand the expert memory from a few demonstrations, and improve the training efficiency of DRL algorithm through gradually unbiased sampling rules. Finally, a compound obstacle avoidance reward system is designed to synchronously realize the collision free movement of the robot end effector and the fuselage, which can build a dense reward mapping and achieve a balance between obstacle avoidance and motion exploration.

3.1 Obstacle avoidance trajectory planning guided by expert network

In this paper, a step-by-step obstacle avoidance strategy is proposed, which aims to cope with the rapid re-planning of changing work scenarios and perform simplified strategy learning for complex obstacles.

Inspired by transfer learning, the Prophet Policy Network Guided Method (PAC) proposed in this paper is a general framework, which can be used for all DRL algorithms based on Actor-Critic framework, such as DDPG, TD3, SAC, etc. This subsection will describe implementation detail of PAC strategy learning proposed in this paper.

PAC is an algorithm framework combining deep reinforcement learning and dynamic strategy optimization, which aims to improve the efficiency and safety of Obstacle Avoidance Trajectory Planning of manipulator in complex environment. This method uses the strategy network to guide the training process, and solves the problems of low exploration efficiency and insufficient sample utilization of traditional reinforcement learning in the sparse reward scene

The core of PAC method is to dynamically adjust the exploration and utilization balance of manipulator through the combination of strategy network and prior knowledge guidance. Specifically, it includes:

(1) Strategy network architecture: adopt the deep deterministic strategy gradient (ddpg) framework to build a dual network structure based on actor critical. The actor network outputs continuous actions (such as joint angle and speed), and the critical network evaluates the action value and guides the strategy update.

(2) Prophet guidance mechanism: dynamic reward function and trajectory prediction module are introduced

to generate auxiliary reward signal through prior knowledge (such as inverse kinematics solution and obstacle position prediction) to accelerate model convergence

Based on the traditional sparse reward (such as successfully avoiding obstacles or reaching the target), the distance penalty (negatively related to the distance of obstacles) and the kinematic reward (based on the minimization of joint power) are added to guide the manipulator to choose a better path.

The proposed PAC consists of the original network and the prophet network, both of which have the same network structure and both include the complete Actor-Critic framework, as shown in Figure 1. The original online Actor network is denoted as $\mu_0\left(s_t\left|\theta^{\mu_0}\right.\right)$ with network parameter θ^{μ_0}, and the prophet online Actor network is denoted as $\mu_p\left(s_t\left|\theta^{\mu_p}\right.\right)$ with network parameter θ^{μ_p}. The original online Critic network is denoted $Q_o\left(s_t,a_t\left|\theta^{Q_o}\right.\right)$ with network parameter θ^{Q_o}, and the prophet online Critic network is denoted $Q_p\left(s_t,a_t\left|\theta^{Q_p}\right.\right)$ with network parameter θ^{Q_p} [25].

At each time step *t*, for state s_t, the original online Actor network and the prophet online Actor network generate the original action a_t^o and the prophet action a_t^p, respectively.

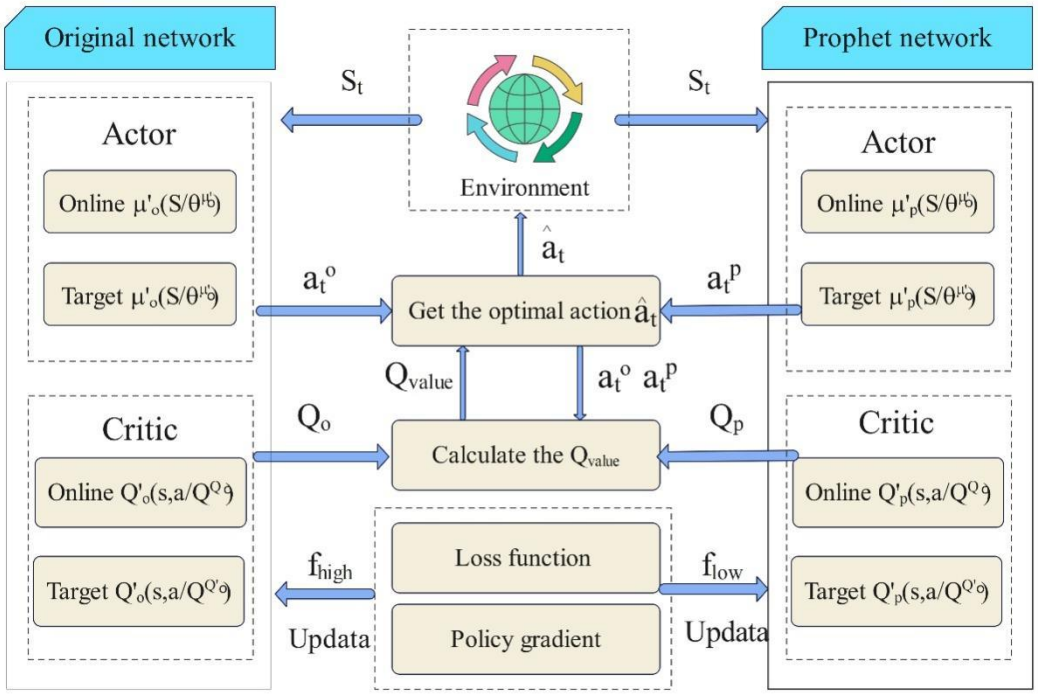

Figure 1: Prophet strategy network framework

$$\begin{cases} a_t^o = \mu_0\left(s_t\left|\theta^{\mu_0}\right.\right) + N_t \\ a_t^p = \mu_p\left(s_t\left|\theta^{\mu_p}\right.\right) + N_t \end{cases} \quad (1)$$

Among them, N_t is random action noise.

The original online Critic network and the prophet online Critic network will jointly comprehensively evaluate the two actions in formula (1).

$$Q_{value}\left(s_t,a_t\right) = \sum_{\theta^{Q_j}\in\left\{\theta^{Q_o},\theta^{Q_p}\right\}} Q\left(s_t,a_t\left|\theta^{Q_j}\right.\right) \quad (2)$$

Then, the action with the highest comprehensive score value $Q_{value}\left(s_t,a_t\right)$ is defined as the optimal action $\hat{a}_t$.

$$\hat{a}_t = arg\,max\,Q_{value}\left(s_t,a_t\right) \quad (3)$$

According to the Bellman equation, the target Q values of the original network and the prophet network Q_{target}, Q_{target} is defined as[26]:

$$Q_{target} = r_t + \gamma Q'\left(s_{t+1},\mu'\left(s_{t+1}\left|\theta^{\mu'}\right.\right)\middle|\theta^{Q'}\right) \quad 4)$$

Among them, γ is the discount factor, the target Actor network is denoted as $\mu'(\cdot)$ with network parameter $\theta^{\mu'}$, the target Critic network is denoted as $Q'(\cdot)$ with network parameter $\theta^{Q'}$, and r_t is the reward value of environmental feedback when the time step is t.

For online Critic networks, the time difference error L_1 is defined as

$$L_1 = \frac{1}{n}\sum_{t=1}^{n}\left(Q_{target} - Q\left(s_t, a_t \middle| \theta^Q\right)\right)^2 \quad (5)$$

In order to embody the guiding ability of the prophet network in the training process of neural network, this paper designs the prophet auxiliary loss function L_2, which is defined as:

$$L_2 = \frac{1}{n}\sum_{t=1}^{n}(1-d_t)\left\|Q^o_{target} - Q^p_{target}\right\|^2 \quad (6)$$

Among them, $\|\cdot\|$ represents the Euclidean norm, u is a Boolean value, and d_t is used to determine whether the TCP reaches the target area within the allowable error range. If TCP successfully reaches the target area, $d_t = 1$ is taken. At this time, $L_2 = 0$ means that the robotic arm of the current round has successfully completed an obstacle avoidance trajectory planning task, so the prophet auxiliary loss function L_2 of the current round is set to zero.

Therefore, the original online Critic network and the prophet online Critic network are updated and optimized by minimizing the composite loss function.

$$L = \lambda_1 L_1 + \lambda_2 L_2 \quad (7)$$

Among them, λ_1 and λ_2 are the weight coefficients, which can be adjusted according to the needs of different obstacle avoidance trajectory planning tasks.

In addition, the original online Actor network and the prophet online Actor network still adopt formula (7) to update the network parameters. Both the target Actor network and the target Critic network adopt the soft update mode of formula (9).

It is worth noting that the formula for updating the Prophet network is the same as the original network. As shown in Figure 2, the Prophet network adopts a delayed update mechanism. For each time step tt, the update frequency of the original network is defined as f_{high}, while the update frequency of the seer network is defined as:

$$f_{low} = \frac{f_{high}}{n_f} \quad (8)$$

Among them, n_f is a positive integer.

3.2 Memory bank design and sampling mechanism

Multi memory database solves the problems of real-time, security and generalization ability of manipulator in complex scenes, and has become the core architecture of intelligent robot control system in the industrial 4.0 era. Its value is particularly prominent in the scenarios of high-speed mixed line production (such as automobile manufacturing) and flexible logistics (such as e-commerce sorting).

The multi memory library can classify and store preset operation procedures (such as palletizing path and welding track) and real-time adjustment instructions, and realize task priority through dynamic scheduling mechanism. For example, in complex sorting scenarios, high priority tasks (such as emergency obstacle avoidance) can immediately call the obstacle avoidance strategy in the memory to interrupt low priority actions The multi memory inventory stores the historical operation parameters (such as grasping force and movement speed), and automatically matches the optimal configuration in similar tasks to reduce the repeated debugging time.

As the basis of deep reinforcement learning algorithm for off-line strategy learning, memory bank has a great impact on the algorithm training. Generally speaking, at the initial stage of training, due to the random exploration of agents, the quality of data stored in the memory is poor, resulting in low learning efficiency. Especially when there are obstacles in the trajectory planning task, it is difficult for robots to obtain high return empirical data. Therefore, the above problems can be solved through multiple memory banks.

According to the amplification mechanism of expert memory bank, this paper constructs a multi-memory bank structure, which aims to amplify a large number of expert memory data by using a small number of artificial expert teaching experience, and adopts a gradual unbiased double memory bank sampling mechanism to improve the exploration efficiency of agents.

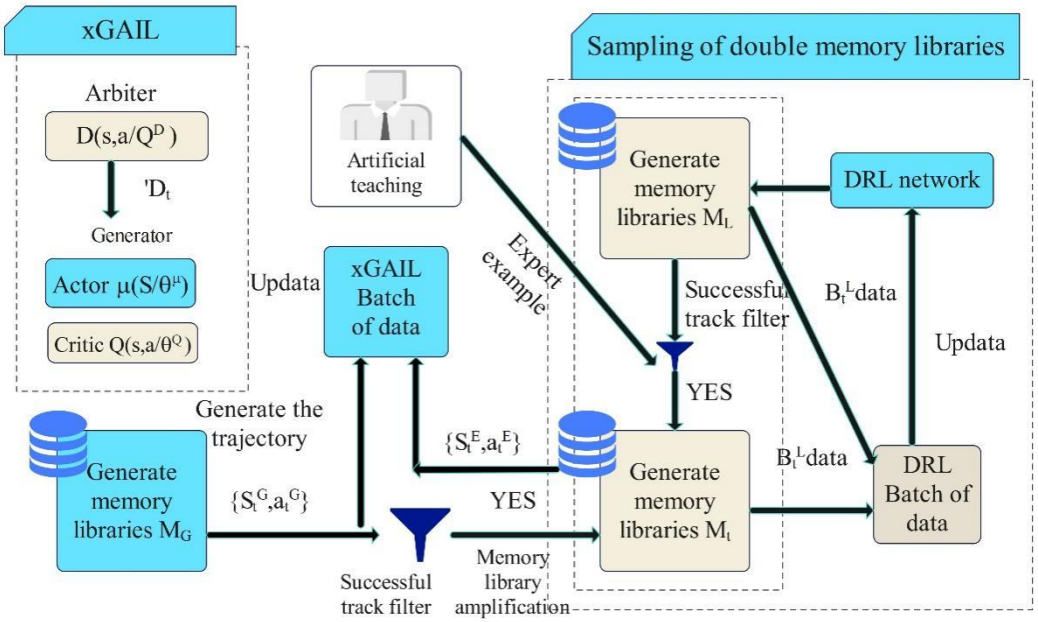

Figure 2: Amplifiable dual memory bank sampling (EDS)

As shown in Figure 2, the dual memory bank sampling method based on memory bank amplification proposed in this paper has two modules that can be parallel: a memory bank amplification module based on xGAIL and a dual memory bank sampling module. Among them, the memory bank amplification module based on xGAIL is mainly used to amplify a small number of expert examples taught manually into a large amount of expert experience data, so as to guide agent learning.

In order to improve the optimization learning efficiency of DRL algorithm strategy, an increasingly unbiased dual-memory sampling mechanism is designed to guide agents' strategy learning. At each time step t, B_t^L and B_t^E data are randomly sampled from the exploration memory bank M_L and the expert memory bank M_E, respectively [27].

$$\begin{cases} B_t^L = \left\lfloor \frac{N}{2} + \frac{t}{T_s} \right\rfloor, B_t^E = N - B_t^L, 0 \le t < \frac{NT_s}{2} \\ B_t^L = N, B_t^E = 0, t \ge \frac{NT_s}{2} \end{cases} \tag{9}$$

Among them, N is the batch data size sampled from the memory bank for each time step t, T_s represents the number of decay steps, and both B_t^L and B_t^E are non-negative integers.

In order to make the sampling more and more unbiased, the value of B_t^E is decreased by 1 every T_s steps, so as to gradually reduce the sampling weight of the data in the expert memory bank M_E. When the DRL converges, B_t^E is set to 0, and its purpose is to allow the DRL to generate empirical data for strategy learning and optimization, so as to achieve an unbiased effect.

3.3 Design of compound obstacle avoidance reward function

Through the compound obstacle avoidance reward function, the collision free motion of the end effector and the fuselage of the robot can be realized synchronously, which can build a dense reward mapping and achieve a balance between obstacle avoidance and motion exploration.

In this section, a compound obstacle avoidance reward system is proposed, which consists of pose error reward function, artificial-like potential field reward function and shortest step reward function, as shown in Figure 3.

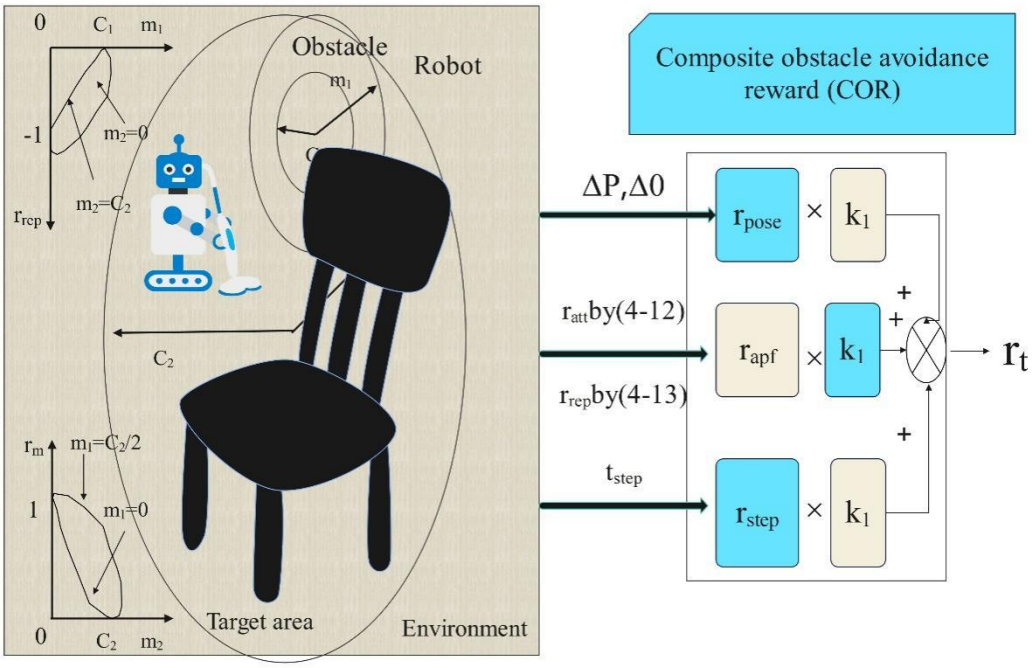

Figure 3: Compound obstacle avoidance reward function system

For the robotic arm trajectory planning task, if the DRL can perceive the completion progress of the task in the entire training stage, the convergence speed of the DRL can be greatly improved, which requires a reward function that can reflect the current task progress of the robotic arm. Therefore, this paper sets the pose error reward function r_{pose} as[28]:

$$r_{pose} = -\|p_1 - p_0\| - \varphi\left(|\alpha_1 - \alpha_0| + |\beta_1 - \beta_0| + |\gamma_1 - \gamma_0|\right) \quad (10)$$

Among them, $p_0 = (x_0, y_0, z_0)$ and $p_1 = (x_1, y_1, z_1)$ are the positions of the robotic arm mission target point and TCP relative to the world coordinate system, $(\alpha_0, \beta_0, \gamma_0)$ and $(\alpha_1, \beta_1, \gamma_1)$ are the attitudes of the robotic arm mission target point and TCP relative to the world coordinate system, and φ is a weight factor used to adjust the importance of position and attitude accuracy to the current task.

Obviously, using r_{pose} as the state reward function of the manipulator can clearly reflect the task progress of the manipulator. When $r_{pose} = 0$, the TCP representing the manipulator completely coincides with the task target point in pose and pose, that is, the manipulator completes the task of reaching the target. In addition, r_{pose} can provide the progress of the robotic arm throughout the training process, which makes the reward more intensive and is conducive to the strategy learning of the robotic arm.

In order to achieve the balance between the two, inspired by the concept of artificial potential field, this paper proposes an obstacle avoidance reward function r_{apf} similar to artificial potential field constructed by quadratic function, which consists of two parts: attraction reward function r_{att} and repulsion reward function r_{rep}, and is defined as:

$$r_{apf} = r_{att} + r_{rep} \quad (11)$$

The influence radius of the obstacle is defined as c_1, which is determined by the safety clearance requirements of the robotic arm for specific tasks. If c_1 is set too small, the manipulator may generate a dangerous trajectory, which is not conducive to practical application. The influence range of the target point is defined as c_2, which is determined by the actual working diameter of the robotic arm, that is, the target point will only have an influence in the working space of the robotic arm. In addition, the minimum three-dimensional spatial distance between the robotic arm and the obstacle in the workspace is defined as m_1, and the three-dimensional spatial distance between the TCP of the jaw at the end of the robotic arm and the target point is defined as j, where

$$m_1 \in \left[0, \frac{c_2}{2}\right] \text{ and } m_2 \in [0, c_2].$$

The attraction reward function r_{att} is defined as a quadratic function passing through the point $(0,1)$ and the point $(c_2, 0)$, designed as:

$$r_{att} = \left(1 - 4c_2^{-1}m_1\right)m_2^2 - \left(c_2^{-1} + c_2 - 4m_1\right)m_2 + 1 \quad (12)$$

Among them, $r_{att} \in [0,1]$ can be inferred from the

function formula, and it can be seen that the closer the distance between TCP and the target point, the greater r_{att}.

The repulsive reward function r_{rep} is a quadratic function through the point $(0,-1)$ and the point $(c_1,0)$, designed as[29]:

$$r_{rep}=\begin{cases}\left(1-2c_2^{-1}m_2\right)m_1^2+\left(c_1^{-1}-c_1+2m_2\right)m_1-1, m_1\le c_1\\ 0, otherwise\end{cases} \tag{13}$$

Among them, $r_{rep}\in[-1,0]$ can be inferred from the function formula, and the farther the minimum three-dimensional spatial distance between the robot and the obstacles in the workspace is, the greater r_{rep} is.

For the practical application of robotic arm, energy optimization is often considered, so as to complete the same task with the least energy consumption, thus reducing the cost. In the training of deep reinforcement learning, reducing the number of steps to complete the same task can also significantly reduce the energy consumption of the robotic arm. Therefore, this paper designs a reward function r_{step} with respect to the number of training steps, which can make the robotic arm reach the target area with as few steps as possible

$$r_{step}=-\left(\frac{t_{step}+1}{N_{step}}\right)+\frac{1}{2} \tag{14}$$

Among them, t_{step} is the number of steps in the current round, and N_{step} is the maximum number of steps in each round. It can be seen that the value range of r_{step} is $\left[-\frac{1}{2},\frac{1}{2}\right]$.

As training progresses, r_{step} will gradually decrease. If the manipulator does not complete the task in the current round, the cumulative value of r_{step} in the whole round is 0. If the manipulator completes the task with only a few steps, the cumulative value of r_{step} in the whole round is larger, which urges the manipulator to learn strategy in the direction of completing the task with a few steps.

Finally, the compound obstacle avoidance reward function (COR) proposed in this paper is defined as:

$$r_t=k_1r_{pose}+k_2r_{apf}+k_3r_{step} \tag{15}$$

Among them, k_1, k_2 and k_3 are the weight coefficients of each reward item.

3.4 Overall implementation plan

As shown in Figure 4, by combining the proposed prophet strategy, amplifiable dual memory bank sampling and compound obstacle avoidance reward function, an inverse kinematics solution framework for obstacle avoidance trajectory planning of robotic arm based on deep reinforcement learning is constructed, and its specific implementation scheme is as follows.

For each step in the formal training, the original online Actor network and the prophet online Actor network output actions a_t^o and a_t^p, respectively from the current state s_t according to formula (1). After passing the calculation Q_{value}, the optimal action $\hat{a}_t$ is obtained by formula (3) to perform the predetermined task. Then, the proposed compound obstacle avoidance reward function is used to obtain reward value feedback, and then the empirical data of the current step is stored in the exploration memory bank M_L. The number N batch data is then sampled by formula (9) through the double memory bank sampling mechanism to update the online network and optimize the target network by soft update. In addition, this paper optimizes the prophet network by delaying update rules.

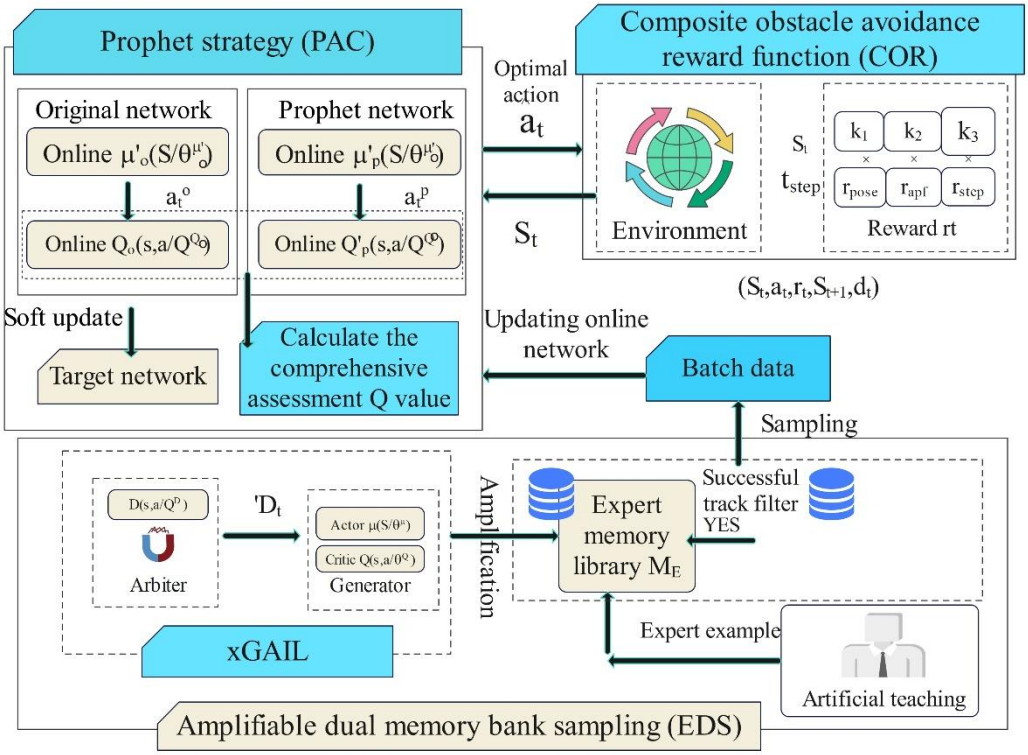

Figure 4: Schematic diagram of overall implementation

During the training process, the successful trajectory filter screens high-quality data from the exploration memory bank M_L to the expert memory bank M_E, and the expert experience data extended by xGAIL will also be screened into the expert memory bank M_E by the generation memory bank M_G. Among them, all memory banks follow the first-in, first-out data flow mechanism.

Finally, the round ends when the TCP of the robotic arm reaches the target area or the number N_{step} of training sessions is reached.

The grasping method based on deep reinforcement learning technology mainly relies on the depth camera to collect data information, transmits the collected image information to the computer for algorithm processing, and then feeds it back to the robotic arm to grasp the information for motion planning. The grasping platform framework is shown in Figure 5, The sensor uses ls-a80020 laser sensor of laser optoelectronics.

The image data input module acquires image data through the camera, which is the visual perception part of the whole system The capture prediction is based on the input image data, and the system predicts the capture position by algorithm. The system outputs the optimal grab position from the grab prediction module, which is the optimal grab point finally determined by the system. The optimal grasping position information determined is transmitted to the control module. The control module is responsible for generating control signals to guide the movement of the manipulator. When the manipulator performs the grasping task, it will feed back its attitude information to the system in real time. This feedback is used to adjust and optimize the motion of the manipulator to ensure accurate execution of the grasping task. The operating system platform is the core control unit of the whole system, which is responsible for integrating and processing all input information (such as image data and manipulator feedback), and coordinating the work of each module.

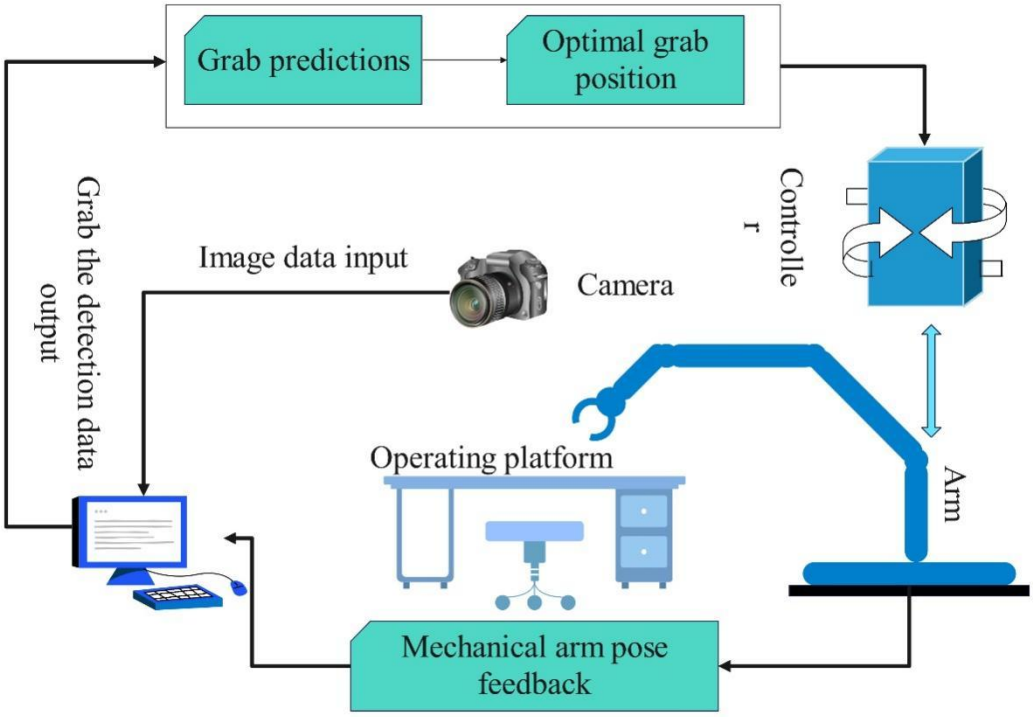

Figure 5: Overall framework of grasping platform

The robotic arm's work of grasping objects is entirely based on visual perception of information, thus realizing autonomous grasping. In the complete grasping detection process, it mainly includes four parts: initialization of simulation environment and detection network, loading of simulation environment and object model, deep enhanced network grasping pose detection, and robotic arm grasping motion. The robotic arm grasping steps are as follows in Figure 6:

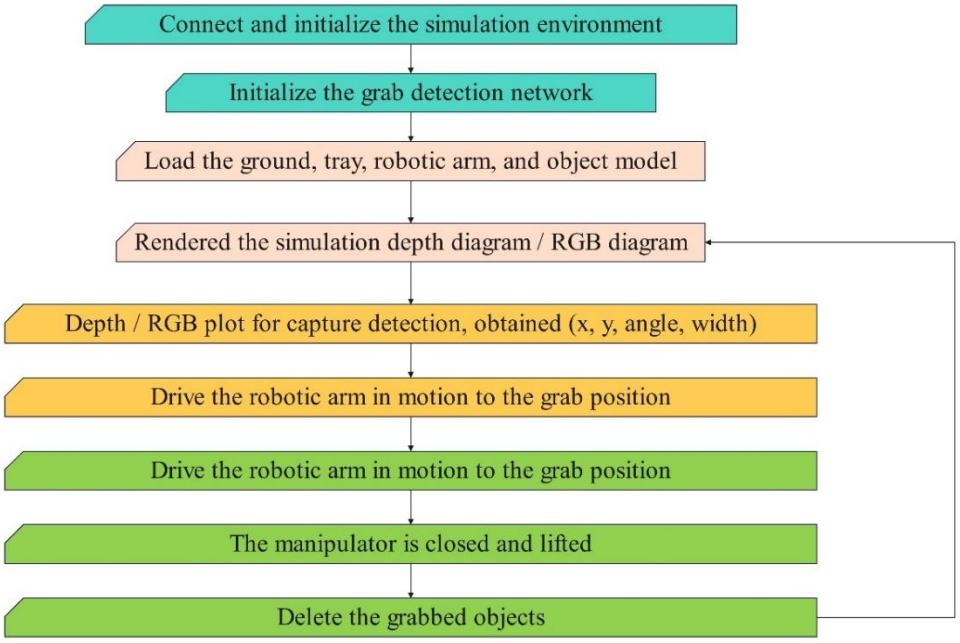

Figure 6: Flowchart of grasping detection

Combining with Figure 6, the grab detection process is analyzed as follows:

In the complete grasping detection process, it mainly includes four parts: initialization of simulation environment and detection network, loading of simulation environment and object model, grasping pose detection of CRE-Net network, and grasping motion of robotic arm. The mechanical arm grasping steps are as follows:

(1) Initialization of simulation environment and detection network: It refers to initializing the Pybullet simulation environment and CRE-Net network.

(2) Loading of simulation environment and object model: it refers to loading models such as ground, tray, robotic arm, grasping object, etc. in Pybullet simulation environment.

(3) Grasping pose detection of CRE-Net network: In the simulation environment, the object model is detected by a Depth camera, and the image is rendered. The collected Depth and RGB images are predicted by CRE-Net network to obtain the key point information (x,y,w,θ) of the grasping pose with the greatest confidence, and the grasping pose is marked on the image, and then the maximum grasping Depth is calculated according to collision detection to provide grasping information for the robotic arm grasping;

(4) Grasping motion of robotic arm: Finally, the grasping pose information is transmitted to the mechanical arm driving mechanism, and the mechanical arm will move to the specified position and pose according to the grasping information to grasp the object. After reaching the specified position, the gripper will open and grasp the object. When the object is successfully grasped, the grasped object model is deleted in the simulation space, and the next round of object grasping detection is carried out.

4 Test

4.1 Test methods and environment

Aiming at the grasping environment of complex multiple unknown objects with messy stacking, this paper proposes an autonomous grasping algorithm of robotic arm based on deep learning, which efficiently solves the problems of easy collision, poor real-time performance and difficulty in capturing reasonable end pose in multi-object stacking scenarios. To improve the performance comparison effect of the model in this article, two scenarios of barrier free and barrier free were set up for comparison, thereby enhancing the intuitive comparison effect of the model The barrier free environment means that there are no other obstacles nearby, only robotic arms and objects to be grasped in environments with obstacles, there are many obstacles in the surrounding environment, which can directly affect the movement path of the robotic arm. Detailed algorithms are needed for path planning to improve the success rate of grasping.

The data set in this paper is a self built data set, which collects tens of thousands of pieces of data through the common objects in life and multiple industrial production parts as the captured objects, and carries out the following comparative test through these data. In addition, in order to further verify the generalization ability of the model, DEX net and YCB data sets are introduced for experiments, with 70% of the data as the training set and 30% of the data as the prediction set.

The Actor network learning rate is set to 0.0003, the Critic network learning rate is set to 0.0003, the soft λ_2 update rate is set to 1, set λ_1 to 1,set N_{step} to 300,set T_S to 3000, c_1 set to 0.05, set c_2 to 1.6, set k_1 to 1, set k_2 to 0.05, set k_3 to 0.1, set to 2000 rounds. set γ to 0.97, set φ to 0.97, set n_f to 0.97.

The model in this article combines images and laser sensors to achieve dual path obstacle avoidance technology, providing reliable reference for the motion path planning of robotic arms and promoting the improvement of gripping efficiency. The trained model is actually tested on the simulated test data set and the data set collected in the real environment, and related ablation experiments and algorithm comparison experiments are carried out. Finally, aiming at the messy stacked multi-object scene, the real robotic arm and the algorithm proposed in this paper are used to grasp tests in a real grasping environment to verify the superiority of this algorithm [30]. The test platform is shown in Figure 7 below.

Figure 7: The test platform

The robot arm model is set to ur10, the camera model is inter realsense d415, the gripper model is z-efg, the processor (CPU) model is Intel Core i9-7900x, and the memory is 128G. NVIDIA GTX 1080ti 2 is selected as the system graphics card, and the memory is 128g.The test hardware equipment is shown in Table 2 below:

Table 2: Test environment

Hardware equipment	Equipment model
Robotic arm model	UR10
Camera model	Inter Realsense D415
Gripping jaw model	Z-EFG
Processor (CPU) model	Intel Core i9-7900X
Processor memory	128G
Graphics card (GPU) model	Nvidia GTX 1080Ti×2
Graphics card memory	128G

A series of software service platforms need to be used when building an autonomous grasping algorithm for robotic arms based on deep reinforcement learning. Among them, the source code and programs of this paper are running under the Linux operating system of Ubuntu16.04. The version of the ROS-based robot operating system is Kinetic. The source code for network construction and part of control uses Python as the main programming language, and the deep learning framework for the network construction part uses the Pytorch framework, which is known for its simplicity and efficiency. Secondly, this paper utilizes two deep learning acceleration libraries, CUDA and Cudnn, which can efficiently perform basic operations of deep neural networks on GPU, such as convolution, pooling, normalization, and activation layers, and can greatly reduce the training and inference time of the network.

REGRAD data set is a multi-dimensional information set containing visual information, grasping information, segmentation information and other information in chaotic stacking scenes. In this paper, the REGRAD capture data set is selected as the training data set and partial test data set for lightweight generative deep reinforcement learning based on capture priority. The REGRAD data set contains sufficient capture information and operational relationship information between multiple objects, which can provide data guarantee for multi-object capture in messy stacking scenarios.

The improved algorithm named AI-PAC-DRL is used for intelligent grasping of two kinds of robotic arms.

SAC-PER and AI-PAC-DRL are experimentally validated. First, we will compare the learning effects and total training time of the two algorithms during the training process to analyze the advantages of the AI-PAC-DRL algorithm. Then, the improvement of MAML algorithm in learning effect and generalization performance is analyzed by testing experiments

4.2 Results

By analyzing the learning curve of the model during the training process, the performance change of the model when the training data is continuously increased can be measured, and the performance of different algorithms can be well compared. Therefore, the SAC-PER algorithm and MAMLSAC algorithm are used to train 10000 rounds in a barrier-free environment, and the obtained learning curve is shown in Figure 8:

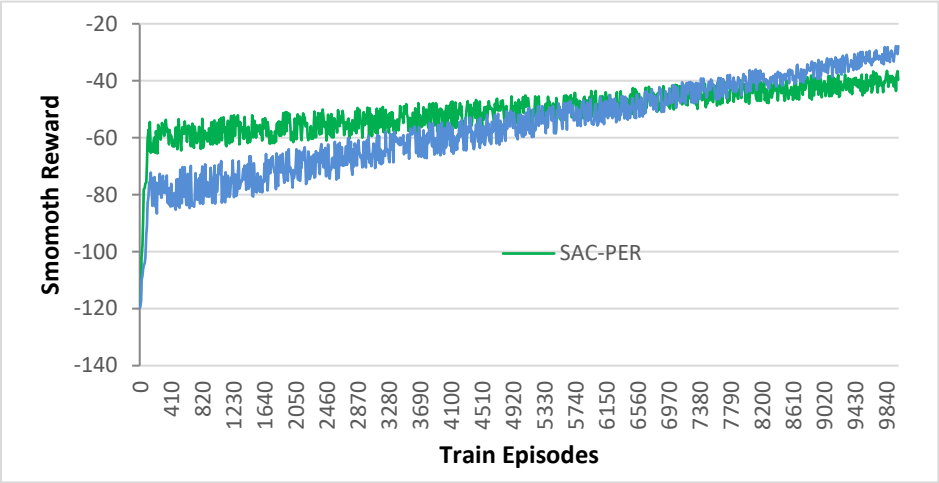

Figure 8: Performance comparison between AI-PAC-DRL algorithm and SAC-PER algorithm in barrier-free environment

In an environment with obstacles, the same setting as without obstacles is adopted, which requires 10,000 rounds of training, and the AI-PAC-DRL algorithm needs to use a base learner to quickly adapt in multiple sub-tasks. The final result of the learning curve is shown in Figure 9:

The trained AI-PAC-DRL algorithm and SAC-PER algorithm are used to conduct 50 tests in eight task scenarios. The final test results are shown in Figure 10:

According to the test, the grasping success rate of each object is shown in Figure 11 and Table 3.

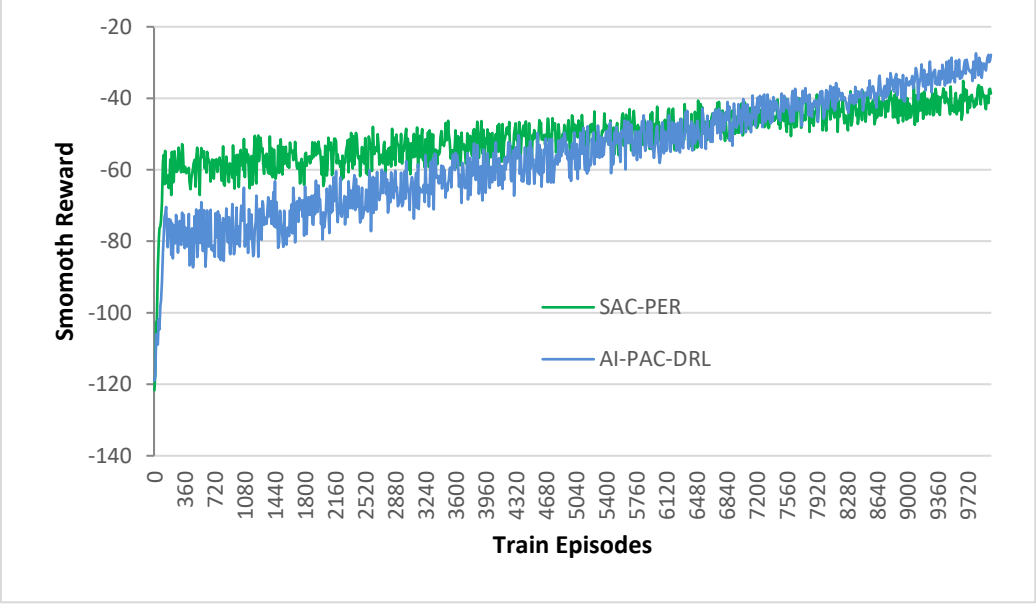

Figure 9: Performance comparison between AI-PAC-DRL algorithm and SAC-PER algorithm in obstacle environment

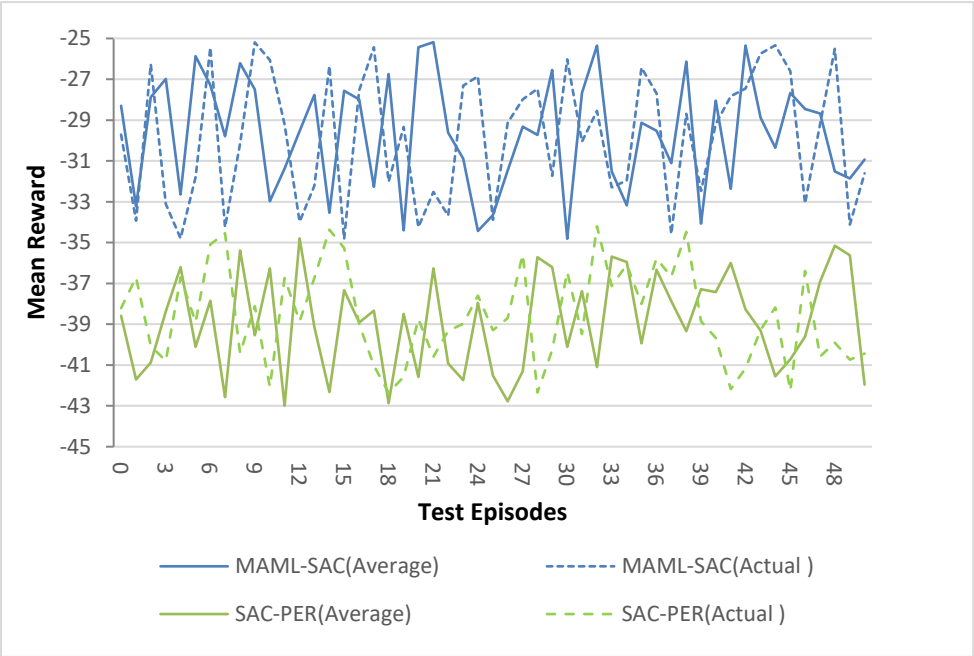

Figure 10: Test results of SAC-PER algorithm and AI-PAC-DRL algorithm in barrier-free environment

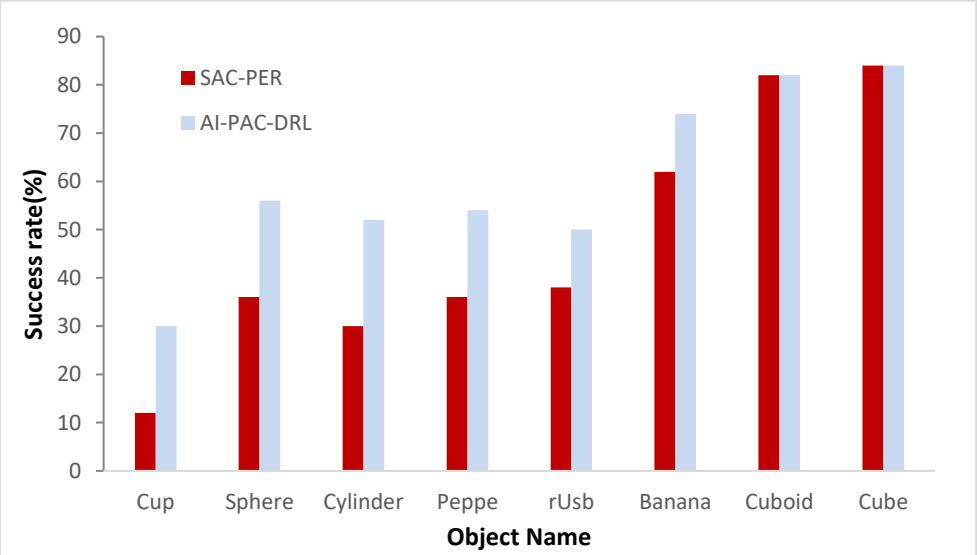

Figure 11: Capture success rate of SAC-PER algorithm and AI-PAC-DRL algorithm in barrier-free environment

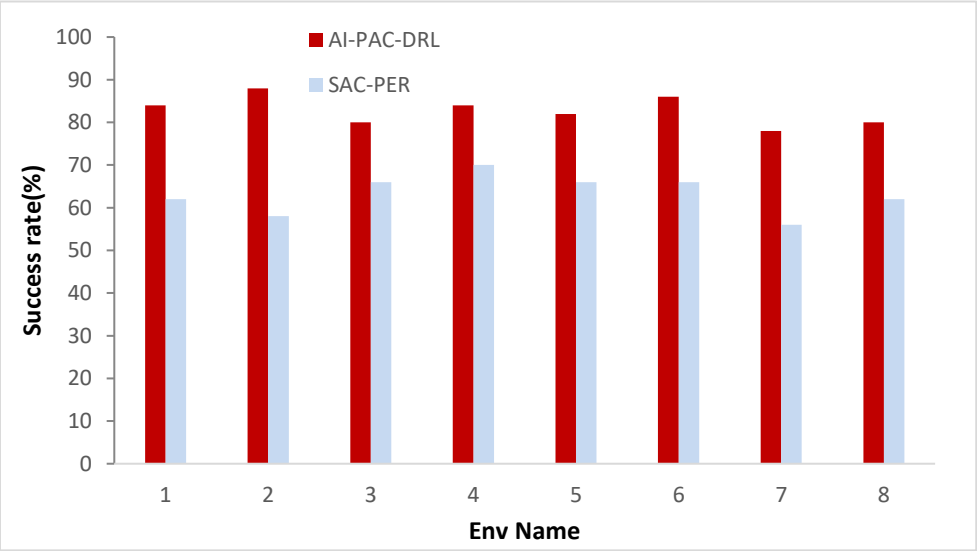

Figure 12: Capture success rate of SAC-PER algorithm and AI-PAC-DRL algorithm in obstacle environment

Table 3: statistical comparison of grab success rate in barrier free environment

	SAC-PER	STD	AI-PAC-DRL	STD	t	P
Cup	12	0.4781	30	0.9749	-3.141	0.00012
Sphere	36	3.2632	56	2.6057	-2.874	0.00017
Cylinder	30	2.2237	52	2.6778	-2.836	0.00032
Peppe	36	3.2369	54	5.1105	-2.654	0.0012
rUsb	38	1.3181	50	3.2562	-2.004	0.0023
Banana	62	5.2019	74	3.8768	-1.212	0.0043
Cuboid	82	6.1623	82	5.9715	0.0021	0.012
Cube	84	2.5647	84	6.9986	0.0011	0.024

After grasping 50 times in each obstacle environment, the test results are shown in Figure 12 and Table 4.

To further verify the complexity of PAC algorithm, the general framework of Robot Obstacle Avoidance Trajectory Planning proposed in this chapter is tested by comparing DRL benchmark algorithms (DDPG, TD3 and SAC) and their corresponding PAC based algorithms (DDPG _pac, TD3_PAC and SAC _ PAC). The test was carried out under the scenario of self built dataset, and Table 5 was obtained.

Table 4: statistical comparison of capture success rate in obstacle environment

	AI-PAC-DRL	STD	SAC-PER	STD	t	P
1	84	4.6119	62	5.7911	-2.921	0.0031
2	88	6.0713	58	5.7865	-3.342	0.00019
3	80	3.7575	66	4.0904	-1.937	0.00065
4	84	5.0530	70	3.4561	-1.532	0.00035
5	82	7.1603	66	4.7836	-0.966	0.00042
6	86	6.3737	66	2.3031	-2.642	0.00064
7	78	5.4021	56	1.9712	-2.024	0.00047
8	80	7.7030	62	2.4861	-1.632	0.00050

Table 5: Complexity analysis of PAC algorithm

Algorithm	Success rate (%)			Training time (Hour)
	1-500	501-1000	1001-2000	
DDPG	6.73	89.50	97.22	24.65
DDPG_PAC	81.77	91.67	98.31	20.79
DDPG_EDS	32.67	97.22	97.71	21.98
DDPG_COR	37.22	99.00	97.81	23.36
DDPG_PEC	89.10	91.08	97.42	20.30
TD3	0.00	70.29	97.02	26.04
TD3_PAC	96.03	95.44	98.70	21.58
TD3_EDS	22.57	98.60	98.41	23.27
TD3_COR	15.84	99.00	98.31	23.66
TD3_PEC	91.87	96.43	98.80	21.29
SAC	3.96	95.24	97.32	24.75
SAC _PAC	95.24	97.02	98.80	20.30
SAC _EDS	40.99	98.60	97.12	23.46
SAC _COR	30.10	98.21	97.81	23.96
SAC _PEC	97.42	96.62	98.60	19.80

The effect of each module on the performance of the model was verified by ablation experiment the benchmark model is the complete model (PAC+EDS+COR), The evaluation indexes are: obstacle avoidance success rate (environmental complexity is divided into low, medium and high levels); Training convergence speed (the number of training steps required to reach 90% success rate); Sample utilization rate (number of effective policy updates per step) The ablation results are shown in Table 6 below.

Table 6: results of model ablation test

Model	Obstacle avoidance success rate (%)			Convergence speed (training steps)	Sample utilization rate (times/thousand steps)
	High	Medium	Low		
PAC + EDS + COR	98.01%	94.05%	87.12%	11880	45
EDS + COR	91.08%	82.17%	69.30%	17820	38
PAC + COR	94.05%	86.13%	74.25%	14850	28
PAC + EDS	84.15%	71.28%	57.42%	19800	41

In order to further verify the generalization effect and practical effect of this model, the algorithm of this model is compared with the existing algorithms, mainly including binocular vision obstacle avoidance technology (bvoa), 3D structured light technology (3D SLT), laser radar technology (3D TOF), visual slam and dynamic path planning (slam-dpp). In the obstacle environment, 3000 rounds of training are carried out to make the model have a certain obstacle avoidance ability. The data set is mainly DEX, YCB data, after that, the model is used to count the success rate of grasping in complex environment, and the test results shown in Table 7 below are obtained.

Table 7: Comparison of model performance under different data sets

Data set	DEX	YCB
BVOA	75.36%	72.52%
3D SLT	80.21%	78.36%
3D TOF	84.36%	83.01%
SLAM-DPP	85.32%	82.35%
AI-PAC-DRL	91.32%	89.35%

4.3 Analysis and discussion

As shown in Figure 8, the blue line represents the learning curve of the AI-PAC-DRL algorithm, the green line represents the learning curve of the SAC-PER algorithm, and the hatched portion represents the actual reward value obtained before the smoothing process. By analyzing the reward curve, it can be seen that the AI-PAC-DRL algorithm has obvious oscillation in the early stage and its learning speed is slower than that of SAC-PER algorithm. The reason is that the AI-PAC-DRL algorithm needs to be trained in multiple sub-task environments in turn in the early stage, and only 125 rounds of training are performed in each sub-task environment. However, after 2000 rounds of meta-reinforcement learning, the AI-PAC-DRL algorithm obtains better initial network parameters through summarizing experience, which can quickly converge to the optimal value along the gradient direction. Therefore, the learning speed is greatly improved, and finally it converges to a higher reward value. Compared with the convergence speed, the AI-PAC-DRL algorithm converges after about 6000 rounds, while the SAC-PER algorithm approaches convergence after about 8000 rounds, which shows that the AI-PAC-DRL algorithm has faster learning speed. Therefore, it shows that the AI-PAC-DRL algorithm has better learning effect than the SAC-PER algorithm

As can be seen from Figure 9, the blue portion represents the reward obtained by the AI-PAC-DRL algorithm, and the green portion represents the reward obtained by the SACPER algorithm. By analyzing the reward curve, it can be seen that in the early stage of training, the convergence speed of AI-PAC-DRL algorithm is relatively slow, and the reward value obtained is lower than that of SAC-PER algorithm. The reason is that the AI-PAC-DRL algorithm learns fewer rounds in each subtask than the SAC-PER. However, with the increase of training rounds, at about 4000 rounds, the AI-PAC-DRL algorithm improves the rapid adaptability of network parameters by summarizing the characteristics of obstacles in sub-task scenarios, making the performance of the algorithm improved, so the reward obtained after the final convergence is higher than that of the SAC-PER algorithm. Then, from the analysis of the overall convergence speed, it can be found that the AI-PAC-DRL algorithm also reaches convergence 2000 rounds earlier than the SACPER algorithm, which shows that the AI-PAC-DRL algorithm has faster learning speed and better performance than the SAC-PER algorithm.

As can be seen from Figure 10, the solid blue line represents the average reward obtained by the AI-PAC-DRL algorithm in the test phase, while the dotted line represents the actual reward obtained by the AI-PAC-DRL algorithm in each environment. The solid green line represents the average reward obtained by the SAC-PER algorithm, and the dotted line represents its actual reward. By comparing the average reward curves of the two algorithms, it can be seen that the AI-PAC-DRL algorithm performs better than the SAC-PER algorithm in both the average reward obtained and the individual reward in each environment. It shows that the AI-PAC-DRL algorithm can summarize the best grasping strategy from objects with different shapes, thus improving the robustness of the algorithm and having higher success rate and reliability.

As can be seen from Figure 11, the red part represents the success rate of the AI-PAC-DRL algorithm test, and the blue part represents the success rate of the SAC-PER algorithm. By analyzing the histogram, it can be seen that the AI-PAC-DRL algorithm has a certain improvement in the success rate of capturing each object. In particular, cup-shaped objects, cylindrical objects, spherical objects, etc., which are quite different in shape from the target objects in the training stage, have better success rates. This shows that the AI-PAC-DRL algorithm has better generalization performance than the SAC-PER algorithm in the obstacle-free environment. In Table 3, from the standard deviation, t value and P value, the data are basically statistically significant, which verifies the effectiveness of the comparative test data.

It can be seen from Figure 12 that the AI-PAC-DRL algorithm has a higher capture success rate in obstacle scenes at different angles. This shows that the meta-learner on AI-PAC-DRL can summarize the characteristics of obstacles well, thus improving the adaptability of the algorithm to new tasks, making the algorithm learn better on the original basis, and obtaining higher success rate. Therefore, it can be concluded that the AI-PAC-DRL algorithm has better generalization performance as well as adaptability to unknown tasks than the SAC-PER algorithm. In Table 4, from the standard deviation, t value and P value, the data are basically statistically significant, which verifies the effectiveness of the comparative test data.

For some failure cases, the main reasons are as follows:

(1) Dynamic obstacles and complex background interference

The trajectory of dynamic obstacles in a chaotic environment is unpredictable, and the background noise (such as irregular object stacking) will interfere with the positioning and segmentation of the target object by the vision system, resulting in the failure of obstacle avoidance path planning or the offset of grasp coordinates. For example, transparent or reflective objects may make the visual sensor unable to accurately capture the depth information, causing error in obstacle avoidance judgment. In addition, some cluttered environments may lead to the manipulator unable to reach the front of the object through an effective path, leading to the inevitable failure of the robot to grasp the object.

(2) Object occlusion and light change when the target object is partially occluded by other objects, the robot cannot obtain complete visual feature information, resulting in the error of grasping attitude estimation; At

the same time, uneven or sudden change of ambient light will reduce the image quality and affect the visual positioning accuracy.

(3) Sensor limitations

It is difficult for a single vision sensor (such as a monocular camera) to achieve high-precision 3D reconstruction in complex scenes, especially when the color of the target object is close to the background, which is easy to lead to the error of obstacle avoidance decision.

In Table 5, the DRL algorithm based on PAC shows faster convergence speed, less average round steps and higher success rate than the benchmark algorithm. Especially in terms of round rewards, Ddpg_pac, td3_pac and sac_pac are all in the early stage of training, and quickly reach a high round reward value, which means that the PAC proposed can provide good expert strategy guidance for the manipulator, so that the manipulator can learn the obstacle avoidance strategy in a very short time.

Each DRL algorithm with the proposed complete PEC framework (that is, PAC, EDS and cor are combined at the same time) maintains a high success rate in the whole training process, and only needs to consume the least training time in almost all scenarios, which shows that the proposed PEC framework has a strong ability in optimizing strategy decision, improving algorithm learning efficiency and enhancing action exploration ability,

In Table 6, after PAC is removed, the success rate of the model in high complexity environment decreases by 18%, and the convergence speed slows down significantly (+6k steps), indicating that PAC improves the stability and generalization of strategy through expert strategy self optimization ability; After removing EDS, the sample utilization rate was reduced by 38%, which verified that the dual memory mechanism optimized the data utilization efficiency through balanced exploration and utilization; The removal of cor resulted in a 14% and 23% reduction in the success rate of low/medium complexity environments, respectively, indicating that cor effectively alleviated the sparse reward problem and enhanced the adaptability to concave polyhedral obstacles through the composite reward function.

In general, PAC is the core component of the model to deal with complex environments through the ability of policy self optimization and scene generalization. The efficient sampling mechanism of EDS significantly improves the training efficiency and reduces the data demand. The compound reward function of cor solves the local optimal and sparse reward problems of the traditional obstacle avoidance model.

In Table 7, AI-PAC-DRL has certain advantages over the existing more advanced models in terms of capture success rate. Compared with SLAM-DPP, which has the best performance in the existing algorithms, AI-PAC-DRL has improved its performance in dex data set by 6% and YCB data set by 7% In general, AI-PAC-DRL outperforms the existing model algorithms in robot path planning and object grasping.

The experimental results show that the improved deep reinforcement learning algorithm has faster convergence speed and learning effect in the training stage, and has higher grasping success rate and generalization energy in the testing stage.

In this paper, the model is trained in a simulated environment, and the capture scene is arranged through the actual environment. From the actual situation, even if a more complex environment is set, the capture accuracy can reach more than 80% after the simulation training The training effect, accuracy and application effect of the model can be further improved through multiple practical operations from this courseware, the model can effectively complete the transition from simulation to reality.

5 Conclusion

This study proposes a collaborative optimization framework based on deep reinforcement learning, which aims to break through the problem of precise grasping of manipulator in complex environment.

The purpose of this paper is to improve the robot's obstacle avoidance effect based on the existing model, and to solve the problem of customer service data training, so as to improve the robot's intelligent control effect and autonomous decision-making ability. Experimental results show that the proposed PAC has the ability of efficient exploration, significantly accelerates the convergence process of the algorithm, reduces the number of training steps, and improves the success rate. While amplifying expert experience data through xGAIL, unbiased dual memory bank sampling rules are used to improve training efficiency. Then, through the increasingly unbiased sampling mechanism, the guiding effect of expert experience data in the early stage of algorithm training is improved, and the influence of expert experience data on the convergence of the algorithm is reduced and gradually eliminated. The experimental results show that the proposed EDS can improve the utilization efficiency of expert experience data, and the success rate of the algorithm is also significantly improved.

Based on the analysis of the failure cases in the experiment, the reasons are obtained, and the research direction of the follow-up paper is obtained. Firstly, the countermeasure training can be introduced to generate a variety of chaotic environment samples to improve the generalization ability of the model; Secondly, hierarchical reinforcement learning (HRL) was used to separate obstacle avoidance and grasping subtasks, or combined with long-term and short-term memory (LSTM) to deal with temporal dependence; Finally, stage rewards (such as bonus points for approaching the target and deduction points for collision) are designed to

balance obstacle avoidance and capturing the target Through the combination of theory and experiment, the model is further verified and improved, and its practical application effect is improved.

Data availability statement

All data generated or analysed during this study are included in this article.

References

[1] Arshad, J., Qaisar, A., Rehman, A. U., Shakir, M., Nazir, M. K., Rehman, A. U., Eldin, E. T., Ghamry, N. A & Hamam, H. (2022). Intelligent control of robotic arm using brain computer interface and artificial intelligence. Applied Sciences, 12(21), 10813-10824. DOI:10.3390/app122110813

[2] Xu, K., & Wang, Z. (2023). The design of a neural network-based adaptive control method for robotic arm trajectory tracking. Neural Computing and Applications, 35(12), 8785-8795. DOI:10.1007/s00521-022-07646-y

[3] Dai, Y., Xiang, C., Zhang, Y., Jiang, Y., Qu, W., & Zhang, Q. (2022). A review of spatial robotic arm trajectory planning. Aerospace, 9(7), 361-372. DOI:10.3390/aerospace9070361

[4] Rawat, D., Gupta, M. K., & Sharma, A. (2023). Intelligent control of robotic manipulators: a comprehensive review. Spatial Information Research, 31(3), 345-357. DOI:10.1007/s41324-022-00500-2

[5] Zaitceva, I., & Andrievsky, B. (2022). Methods of intelligent control in mechatronics and robotic engineering: A survey. Electronics, 11(15), 2443-2454. DOI :10.3390/electronics11152443

[6] Abdi, A., Ranjbar, M. H., & Park, J. H. (2022). Computer vision-based path planning for robot arms in three-dimensional workspaces using Q-learning and neural networks. Sensors, 22(5), 1697-1705. DOI :10.3390/s22051697

[7] Ai, J., Meng, J., Mai, X., & Zhu, X. (2023). BCI control of a robotic arm based on ssvep with moving stimuli for reach and grasp tasks. IEEE Journal of Biomedical and Health Informatics, 27(8), 3818-3829. DOI: 10.1109/JBHI.2023.3277612

[8] Mourtzis, D., Angelopoulos, J., & Panopoulos, N. (2022). Closed-loop robotic arm manipulation based on mixed reality. Applied Sciences, 12(6), 2972-2982. DOI:10.3390/app12062972

[9] Zhou, Y., Xie, L., & Pan, H. (2022). Research on a PSO-H-SVM-based intrusion detection method for industrial robotic arms. Applied Sciences, 12(6), 2765-2773. DOI:10.3390/app12062765

[10] Singh, B., Kumar, R., & Singh, V. P. (2022). Reinforcement learning in robotic applications: a comprehensive survey. Artificial Intelligence Review, 55(2), 945-990. DOI:10.1007/s10462-021-09997-9

[11] Abdullah-Al-Noman, M., Eva, A. N., Yeahyea, T. B., & Khan, R. (2022). Computer vision-based robotic arm for object color, shape, and size detection. Journal of Robotics and Control (JRC), 3(2), 180-186. DOI :10.18196/jrc. v3i2.13906

[12] Xu, B., Li, W., Liu, D., Zhang, K., Miao, M., Xu, G., & Song, A. (2022). Continuous hybrid BCI control for robotic arm using noninvasive electroencephalogram, computer vision, and eye tracking. Mathematics, 10(4), 618-624. DOI :10.3390/math10040618

[13] Liu, Z., Peng, K., Han, L., & Guan, S. (2023). Modeling and control of robotic manipulators based on artificial neural networks: a review. Iranian Journal of Science and Technology, Transactions of Mechanical Engineering, 47(4), 1307-1347. DOI:10.1007/s40997-023-00596-3

[14] Tang, Z., Wang, P., Xin, W., & Laschi, C. (2022). Learning-based approach for a soft assistive robotic arm to achieve simultaneous position and force control. IEEE Robotics and Automation Letters, 7(3), 8315-8322. DOI: 10.1109/LRA.2022.3185786

[15] Ahmed, A., Yu, M., & Chen, F. (2022). Inverse kinematic solution of 6-DOF robot-arm based on dual quaternions and axis invariant methods. Arabian Journal for Science and Engineering, 47(12), 15915-15930. DOI: 10.1007/s13369-022-06794-6

[16] Kim, E., Shin, J., Kwon, Y., & Park, B. (2023). EMG-based dynamic hand gesture recognition using edge AI for human–robot interaction. Electronics, 12(7), 1541-1550. DOI:10.3390/electronics12071541

[17] Wang, C., Li, C., Han, Q., Wu, F., & Zou, X. (2023). A performance analysis of a litchi picking robot system for actively removing obstructions, using an artificial intelligence algorithm. Agronomy, 13(11), 2795-2803. DOI:10.3390/agronomy13112795

[18] Tran, D. T., Vo, H. Q., Nguyen, T. K., & Nguyen, T. N. (2025). Enhanced teleoperation and visual-force feedback with obstacle avoidance for a car-like mobile robot based on WAN network architecture. Journal of Technical Education Science, 20(01), 62-72. DOI :10.54644/jte.2025.1601

[19] Wu, P., Su, H., Dong, H., Liu, T., Li, M., & Chen, Z. (2025). An obstacle avoidance method for robotic arm based on reinforcement learning. Industrial Robot: the international journal of robotics research and application, 52(1), 9-17. DOI: 10.1108/IR-05-2024-0206

[20] Ricárdez Ortigosa, A., Bestmann, M., Heilemann, F., Halbe, J., Christiansen, L., Rodeck, R., & Wende, G. (2025). Foundations for teleoperation and motion planning towards robot-assisted aircraft

fuel tank inspection. Aerospace, 12(2), 156. DOI :10.3390/aerospace12020156

[21] Raj, R., & Kos, A. (2025). An extensive study of convolutional neural networks: applications in computer vision for improved robotics perceptions. Sensors, 25(4), 1033. DOI: 10.3390/s25041033

[22] Ušinskis, V., Nowicki, M., Dzedzickis, A., & Bučinskas, V. (2025). Sensor-fusion based navigation for autonomous mobile robot. Sensors, 25(4), 1248. DOI :10.3390/s25041248

[23] Wang, D., Liu, B., Jiang, H., & Liu, P. (2025). Path planning for construction robot based on the improved a* algorithm and building information modeling. Buildings, 15(5), 719. DOI: 10.3390/buildings15050719

[24] Zhu, W., Gao, X., Wu, H., Chen, J., Zhou, X., & Zhou, Z. (2025). Design of multimodal obstacle avoidance algorithm based on deep reinforcement learning. Electronics, 14(1), 78. DOI:10.3390/electronics14010078

[25] Zhang, C., Zhang, X., Yang, W., Zhang, G., Wan, J., Lei, M., & Dong, Z. (2025). Safe path planning method based on collision prediction for robotic roadheader in narrow tunnels. Mathematics, 13(3), 522. DOI: 10.3390/math13030522

[26] Bengueddoudj, A., Belhadj, F., Hu, Y., Zitouni, B., Idir, Y., Adoui, I., & Mostefai, M. (2025). Efficient line-based visual marker system design with occlusion resilience. *Informatica*, *49*(1). DOI: 10.31449/inf. v49i1.7259

[27] Aradea, A., Rianto, R., Herlina, N., & Hoeronis, I. (2025). Self-learning model for pattern recognition in vision system based on adaptive kernel. Informatica, 49(14). DOI: 10.31449/inf. v49i14.7272

[28] Qian, P. (2025). Dual-layer dynamic path optimization for airport ground equipment using graph theory and adaptive genetic algorithms. Informatica, 49(13). DOI: 10.31449/inf. v49i13.7651

[29] He, X., Wang, R., Cao, T., Liang, W., & Fan, Y. (2025). Fusion CNN-transformer model for target counting in complex scenarios. Informatica, 49(12). DOI: https: 10.31449/inf. v49i12.7315

[30] Abdillah, M. S., Masykur, F. and Sasmowiyono, S. S. (2025). Design and construction of a smart wheelchair from a lecture chair using power window motors with smartphone control for disability. International Journal of Engineering, 38(6), 1397-1404. DOI: 10.5829/ije.2025.38.06c.12