Article | Received 27 November 2025; Revised 30 April 2026; Accepted 9 May 2026; Published 21 May 2026
https://doi.org/10.55092/rl20260017

Robot assembly strategy optimization based on embodied skill learning

Fengming Li[1], Hui Qi[1], Ligang Jin[2,*], Xiaoqing Yao[3], Yu Men[3] and Rui Song[3]

[1] School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan, China
[2] School of Artificial Intelligence, Shandong University, Jinan, China
[3] School of Control Science and Engineering, Shandong University, Jinan, China

* Correspondence author; E-mail: lgjin@sdu.edu.cn.

Highlights:

- A search strategy that integrates force and position information has been proposed.
- A phased assembly strategy incorporating reinforcement learning is used to achieve the entire assembly task.
- Fuzzy reward systems is proposed to accelerate the assembly process.

Abstract: Robotic assembly is a crucial component of intelligent manufacturing, significantly enhancing the level of automation in the industry. Traditional robotic control strategies struggle to adapt to complex and dynamic industrial environments. Learning-based control methods, which incorporate perception, decision, and planning, can greatly improve the adaptability of assembly strategies. This paper addresses the diverse assembly production demands in dynamic operational scenarios and takes a deep dive into the mechanical characteristics at different stages of peg-in-hole assembly. Our research focuses mainly on the construction of compliant robotic embodied assembly strategy models and the acquisition of staged assembly skills, addressing the complexities of strategy models and the low efficiency of robotic skill learning during the peg-in-hole assembly process. Firstly, the peg-in-hole assembly task is divided into distinct stages, and a compliant force control method is proposed based on mechanical characteristics. Subsequently, a robotic assembly strategy model based on proximal policy optimization (PPO) is introduced, combined with a fuzzy quality evaluation system to achieve staged acquisition and optimization of the robotic compliant assembly strategy under the embodied strategy learning framework. Finally, experimental validation was conducted in a simulated peg-in-hole assembly environment. The success rate of this algorithm consistently remains above 98%, representing a 12% improvement over the deep deterministic policy gradient algorithm. The results confirm that the proposed embodied strategy offers an effective solution for acquiring sophisticated assembly skills in uncertain environments.

Keywords: robotic assembly; deep reinforcement learning; compliant control; fuzzy logic

Copyright©2026 by the authors. Published by ELSP. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium provided the original work is properly cited.

1. Introduction

Assembly, as the last process of manufacturing, affects the final quality of the product [1]. Therefore, the realization of intelligent manufacturing is inherently dependent on the successful implementation of intelligent assembly systems [2]. As a critical enabling technology for intelligent manufacturing [3], industrial robots have gained widespread applications in precision machining, welding, and assembly operations, owing to their characteristics of high precision and superior payload capacity [4]. Traditional robotic control strategies struggle to adapt to complex and dynamic industrial environments. Learning-based control methods, which incorporate perception, decision, and planning, can greatly improve the adaptability of assembly strategies. This paper addresses the diverse assembly production demands in dynamic operational scenarios and takes a deep dive into the mechanical characteristics at different stages of peg-in-hole assembly.

As a core direction of intelligent robotics, embodied intelligence emphasizes that adaptive manipulation skills originate from the dynamic interaction between the robot's physical body and the environment [5]. Focusing on the diverse production demands of dynamic peg-in-hole assembly, this paper deeply analyzes the mechanical characteristics of the search and insertion stages, and aims to solve the problems of complex strategy modeling and low skill learning efficiency in robotic peg-in-hole assembly. We construct a compliant embodied assembly strategy model based on proximal policy optimization (PPO) algorithm, integrate force-position-vision information and fuzzy logic reward system, and realize the staged acquisition and optimization of assembly skills. Experimental validation in both simulated and real environments verifies that the proposed strategy significantly improves the assembly success rate and efficiency in uncertain environments. The main contributions are as follows:

(1) A comprehensive workflow for peg-in-hole assembly operations has been designed. In the initial stage, random deviations between the peg and the hole are set. A tri-modal perception fusion search strategy (integrating image, force, and position information) is employed to achieve autonomous and rapid hole localization, breaking through the limitations of single/dual-modal approaches. Once the search is deemed successful by evaluating both force and position information, the insertion strategy is executed to insert the peg into the hole, thereby completing the assembly process.

(2) A method of obtaining the strategy of peg-in-hole assembly by stages is proposed. Based on phase-based task modeling (dividing the assembly task into search and insertion phases according to mechanical differences), the PPO is used as the basic framework to deeply collaborate with Proportional-Integral (PI) compliant force control (resolving the frequency coordination issue to enable compliant assembly with low contact force and high adaptability), learning each phase independently as a separate Markov decision process (MDP). This phase-wise training simplifies modeling complexity, improves learning efficiency, and ultimately enhances both assembly efficiency and safety.

(3) A multi-modal perception search strategy integrating image, force and position information is proposed. This strategy breaks through the limitations of single/dual-modal approaches, avoiding the drawbacks of single visual or force-based search methods and improving the stability of hole

positioning. Complemented by the phase-based fuzzy logic reward system (replacing deterministic rewards with a multi-factor fuzzy evaluation), it enhances the agent's environmental exploration capability, addressing the issues of slow reward convergence and weak guidance in traditional reward design.

The rest of this paper is organized as follows: Section 2 reviews the related work in robotic peg-in-hole assembly, compliant force control and reinforcement learning-based assembly strategies. Section 3 provides a detailed explanation of the methods used in this study. Section 4 presents the experimental validation and result analysis. Section 5 concludes the work of this paper.

2. Related work

Peg-in-hole assembly, as a critical component of robotic assembly [6], is commonly encountered in various industrial manufacturing scenarios such as automotive production, aerospace, and electronics assembly [7]. As a contact-rich task, peg-in-hole assembly can be divided into two stages based on the contact state: the search stage and the insertions stage.

2.1. Peg-in-hole assembly stage control

In the search stage, the setting of search strategy is related to many factors such as execution time, assembly accuracy, stability, shape and material of the workpieces. The search method based on visual servo is one of the key research directions for the peg-in-hole search stage. Xu *et al.* [8] used binocular vision to estimate the pose of the target hole, adaptively adjusting the pose of a monocular camera mounted on the end-effector for visual servoing, and implemented hole positioning through a directional error compensation algorithm. Wu *et al.* [9] utilized a vision-based approach to monitor the state and deformation of the cable in real time. Zhang *et al.* [10] combined intelligent detection algorithms with traditional image processing techniques using a handheld depth camera to perform 6D pose detection of connectors, significantly reducing assembly time. However, vision-based search methods are highly susceptible to issues such as image occlusion and camera accuracy, making precise hole positioning difficult. Li *et al.* [11] proposed a tactile-visual fusion framework for robotic assembly, verifying that integrating force and visual data can reduce positioning errors caused by occlusion or low camera accuracy—this provides direct support for our decision to integrate image, force, and position information in the search stage. Ahn *et al.* [12] proposed an assembly strategy of force-visual fusion can handle large position/orientation errors to complete assembly. Therefore, it is also an important method to obtain hole search strategies by presetting search trajectories based on robot position and force information. In peg-in-hole assembly,commonly used search trajectory include Archimedes spiral curve, square helix and windmill search track [13]. Park *et al.* [14] proposed a search method based on spiral force locus (SFT), but the time required for different search tasks varied greatly. Therefore, they proposed a partial spiral force locus (PSFT) search algorithm to reduce the time variance. Lee *et al.* [15] estimated the contact state of peg-in-hole assembly based on fuzzy logic, and selected appropriate assembly parameters according to the estimated contact state. Although the hole search can be well realized by presetting the search trajectory, the assembly scenario with uncertain search scope takes a long time. How to reduce the waste of time in the search process has emerged as a critical research focus in the field. To address the challenges in

the aforementioned research, we propose a search algorithm that integrates image, force, and position information. This approach does not require object-specific modeling or manual adjustment of assembly parameters. Instead, through training, the peg can achieve autonomous and safe searching in a relatively short time, thereby enhancing search efficiency.

In the insertion stage, the contact information between pegs and the holes is more complex. When there is a misalignment between the relative poses of the peg and hole, large contact force will be generated, which may damage the workpieces and pose a threat to the safety of the robot and the operator. Applying compliant force control in peg-in-hole assembly can avoid large contact force [16]. Hou *et al.* [17] combined with the fuzzy logic system, predicted the assembly environment based on variable time scale prediction to reduce unnecessary exploration by agents. This model uses fuzzy logic to determine the peg-hole contact state and adjusts admittance control parameters accordingly for precise pose estimation and fine-tuning. However, with the development of personalized and customized demands, the same production line may need to assemble different objects or different poses of the same object. This randomness in object types or poses is referred to as dynamic scenarios, where traditional control methods are difficult to adapt. Kim *et al.* [18] proposed an adaptive robotic assembly system empowered by embodied intelligence, which enables robots to adjust clamping force and insertion paths in real time based on workpiece material and dimension deviations. To address the issues in the aforementioned research, we integrate compliant force control with reinforcement learning algorithms. By training under the guidance of force control, we obtain the optimal assembly strategy, achieving the assembly task with minimal force and fewer steps.

2.2. Compliant force control for robotic assembly

Compliant force control is the key technology to realize safe and stable peg-in-hole assembly. Common compliant control methods include impedance control, admittance control and PI force control. Zhao *et al.* [19] achieved compliant interaction between the robot end-effector and the environment through an impedance controller, which also laid the foundation for migration across different tasks. He *et al.* [20] adopted adaptive fuzzy neural network control method to adjust impedance parameters to complete compliant operation. Yang *et al.* [21] extended the impedance parameters to the action space of reinforcement learning, and obtained the peg-in-hole assembly strategy according to the demonstration assembly trajectory.

Fuzzy logic is often integrated into compliant control to improve adaptability. Ning *et al.* [22] employed a multi-layer perceptron for coarse adjustment of peg-hole poses and subsequently proposed a fuzzy variable admittance control model. This model uses fuzzy logic to determine the peg-hole contact state and adjusts admittance control parameters accordingly for precise pose estimation and fine-tuning. Zhang *et al.* [23] established a jamming model for peg-in-hole assembly to analyze the relationship between stuck state and the underlying force controller, which can effectively avoid jamming and successfully complete the assembly. However, most compliant control methods are used independently with fixed parameters, and lack effective combination with learning-based assembly strategies, which limits the further improvement of assembly efficiency in dynamic environments.

2.3. Reinforcement learning-based robotic assembly strategies

Reinforcement learning (RL) is the mainstream method for robotic skill learning in complex environments [24]. Lillicrap *et al.* [25] proposed the Deep Deterministic Policy Gradient (DDPG) algorithm based on the Actor-Critic (AC) framework [26] and applied it to robot grasping tasks in the Multi-Joint dynamics with Contact (MuJoCo) simulation environment. However, the DDPG algorithm requires training multiple networks, so Gu *et al.* [27] introduced the Normalized Advantage Function (NAF), applying the Q-learning algorithm to continuous action spaces. They further improved NAF and successfully applied it to complex tasks such as door opening [28]. PPO algorithm as a classic policy gradient method, has the advantages of stable training and fast convergence, and is gradually applied to robotic assembly tasks. Zou *et al.* [29] designed a fuzzy Q-learning-based variable impedance controller to learn compliant behavior during the robot peg-in-hole assembly process without considering complex physical contact models. Ma *et al.* [30] developed a reinforcement learning framework for shaft sleeve assembly using the Actor-Critic structure and proposed a comprehensive reward function to achieve interference fit assembly. Lee *et al.* [31] and Schoettler *et al.* [32] utilized visual information as network input to accomplish tasks such as peg-in-hole assembly and electrical connector insertion/removal.

Hierarchical reinforcement learning and meta-reinforcement learning are also hot research directions. Hou *et al.* [33] employed hierarchical reinforcement learning to decompose the robot assembly strategy, enabling dual peg-in-hole assembly while improving data utilization efficiency. Liu *et al.* [34] proposed a knowledge rule-guided robot learning method combined with a predictive model. Building upon Cartesian compliant control, they established a knowledge-guided exploration strategy using fuzzy logic based on position/force feedback. This strategy provides exploration direction and limits the exploration range during the early stages of learning. Yan *et al.* [35] proposed an adaptive meta policy learning algorithm, integrating meta-reinforcement learning methods to obtain a generalized model for multi-category assembly skills, enabling multi-peg-in-hole assembly.

Inspired by the above research, we proposed a skill-learning-based method for robotic peg-in-hole assembly, focusing on strategy learning for both the search and insertion stages. The PPO algorithm is used as the basic framework, and the search and insertion stage reward function is designed based on fuzzy logic. This approach is integrated with compliant force control algorithms to ensure the safe and efficient completion of the assembly tasks.

3. Proposed method

3.1. Robotic compliant force control

In order to ensure the compliance and safety during the peg-in-hole assembly, the strategies in the search and insertion stages are completed by the robot's underlying force controller. The force control algorithm is the core control algorithm of the peg-in-hole assembly strategy. To complete the peg-in-hole assembly, the PI control algorithm is used to realize the compliant force control, which can accelerate the speed of assembly and at the same time, can enhance the compliance of the robot, and can effectively prevent the generation of large contact force.

As shown in Figure 1, the assembly action P_t^a interacting with the assembly environment is composed of the assembly displacement output P_t^h by the strategy model and the compliant displacement P_t^g by the force controller. However, due to the different frequency between the strategy model and the force controller, there exists the problem of cooperative control between the strategy action and the compliant displacement output from the force control. Therefore, the assembly displacement P_t^h is obtained by interpolating the strategic action of each step:

$$P_t^h = eP_t, \tag{1}$$

where e is the frequency ratio between the strategy model and the force controller, P_t is the robotic action. The compliant displacement P_t^g output from the force controller is:

$$P_t^g = K_p\Gamma_t^e + \int K_i\Gamma_t^e. \tag{2}$$

where K_p is proportional coefficient matrix, K_i is integral coefficient matrix, both of which include force control parameters in six directions. Γ_t^e is the deviation between the true force and the predefined desired force at time t. The motion interpolation enables the effective assembly displacement P_t^h and the compliant displacement P_t^g output from the force controller at the same frequency, thus realizing cooperative control.

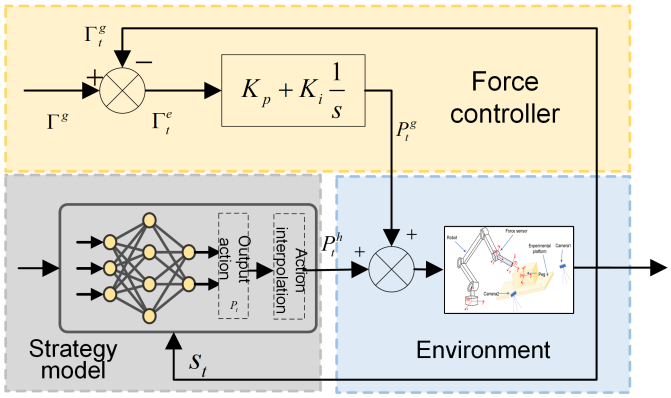

Figure 1. Schematic diagram of the force controller.

3.2. Fuzzy reward system design

3.2.1. Quality evaluation system based on fuzzy logic

Peg-in-hole assembly, as a complex task in industrial scenarios, involves high-dimensional continuous state-action features, which must be considered in reward function design. Compared to deterministic quality evaluation, using fuzzy logic to construct a quality evaluation system can take more learning-promoting factors into account. Additionally, the fuzzy reward function inherently enhances the agent's exploration during the learning process.

Fuzzy logic deals with fuzzy sets on the basis of multi-valued logic, and a complete fuzzy controller consists of components such as fuzzification, rule base, fuzzy inference, and defuzzification. Based on the fuzzy control system, a fuzzy quality evaluation system is constructed as shown in Figure 2, using it as the framework for setting the reward function in strategy learning. The input to the quality evaluation system is the environmental state, which includes sensor-measured environmental data and the robot's positional information, while the output is the reward value needed for the agent's update. The quality evaluation

system consists of two parts: the function-based continuous reward r_1 and the fuzzy logic-based fuzzy reward r_2. The relationship between the two can be expressed as follows:

$$r_t = \omega_1 r_1 + \omega_2 r_2. \tag{3}$$

where r_t is the final reward value, with ω_1 and ω_2 representing the weights of each part respectively.

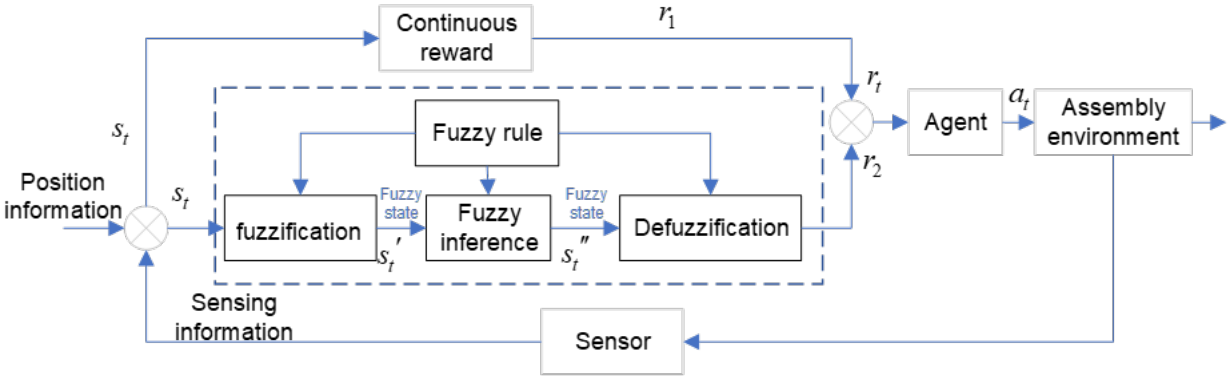

Figure 2. Fuzzy quality evaluation system.

3.2.2. Fuzzy quality evaluation in search stage

The reward function in the search stage consists of three parts: reward r_{se}^1 for increasing search speed, reward r_{se}^2 constructed by fuzzy control, and a critical reward r_{se}^3 for search success/failure:

$$r_{se} = \omega_{se}^1 r_{se}^1 + \omega_{se}^2 r_{se}^2 + \omega_{se}^3 r_{se}^3, \tag{4}$$

where r_{se} is the final reward value, with ω_{se}^1, ω_{se}^2 and ω_{se}^3 representing the weights of each part, with values of 4, 12 and 4, respectively. The setting of reward r_{se}^1 is based on the assembly steps k_{se} in the search stage:

$$r_{se}^1 = -\frac{k_{se}}{k_{se}^m}, \tag{5}$$

where k_{se}^m represents the maximum number of assembly steps allowed in the insertion stage, which is used to limit the maximum assembly time. r_{se}^1 ranges within $[-1,0]$. Reward r_{se}^2 is designed through the fuzzy control system, using the distance between the centers of the peg and the hole as parameter:

$$d_x = P_x^t - P_x^h, \tag{6}$$

$$d_y = P_y^t - P_y^h. \tag{7}$$

where d_x and d_y represent the distance deviations of the peg and hole in the x-axis and y-axis, respectively. P_x^t and P_y^t denote the x-axis and y-axis coordinates of the robot's end effector at time t. P_x^h and P_y^h represent the x-axis and y-axis coordinates of the robot's end effector when it is directly above the plane of the hole. In Figure 3a represents the fuzzy reward system for the search phase. The required parameters d_x and d_y are preprocessed and then input into the fuzzy control system. The triangular membership function is used for fuzzification, and the parameters of the membership functions are shown in Table 1.

Table 1. Membership function parameter.

Input	c-a	c	Domain	Number of Fuzzy Sets
x-direction distance deviation d_x	10	{0,5,10,15,20}	[0,20]	5
y-direction distance deviation d_y	1.25	{0,1.25,2.5,3.75,5}	[0,5]	5

After fuzzifying each parameter, the operation is performed based on the established fuzzy rules and fuzzy inference. The fuzzy rules are established as shown in Table 2.

Table 2. Fuzzy rule base.

d_y \ d_x	VG	G	M	B	VB
VG	0.1	0.2	0.3	0.4	0.5
G	0.2	0.3	0.4	0.5	0.6
M	0.3	0.4	0.5	0.6	0.7
B	0.4	0.5	0.6	0.7	0.8
VB	0.5	0.6	0.7	0.8	0.9

After fuzzifying each parameter, the operation is performed based on the established fuzzy rules and fuzzy inference to obtain the output fuzzy set. Finally, the centroid method is used to defuzzify the fuzzy set into a clear fuzzy reward. The fuzzy control for the search phase is implemented by a single-layer fuzzy controller, with its specific structure illustrated in Figure 3. In the search phase, the inputs are d_x and d_y, and the output is r_{se}. The specific structure of the fuzzy controller in the search phase is shown in Figure 3a. r_{se}^2 ranges within [0,1].

The fuzzification is completed using a triangular membership function, as shown below:

$$f(x) = \begin{cases} 0, & (x \le a) \\ (x-a)/(b-a), & (a \le x \le b) \\ (c-x)/(b-a), & (b \le x \le c) \\ 0, & (x \ge c) \end{cases}, \tag{8}$$

where a, b, and $c$represent the parameter values in the triangular membership function. a and c determine the width of the function value, and b specifies its position. Following fuzzy inference through the constructed rule base, defuzzification is performed using the center-of-gravity method, ultimately yielding the reward value:

$$r_{se}^2 = \frac{\sum_j C_j w_j}{\sum_j C_j}, \tag{9}$$

where C_j is the fuzzy set obtained from fuzzy inference, w_j refers to the weight of each fuzzy set.

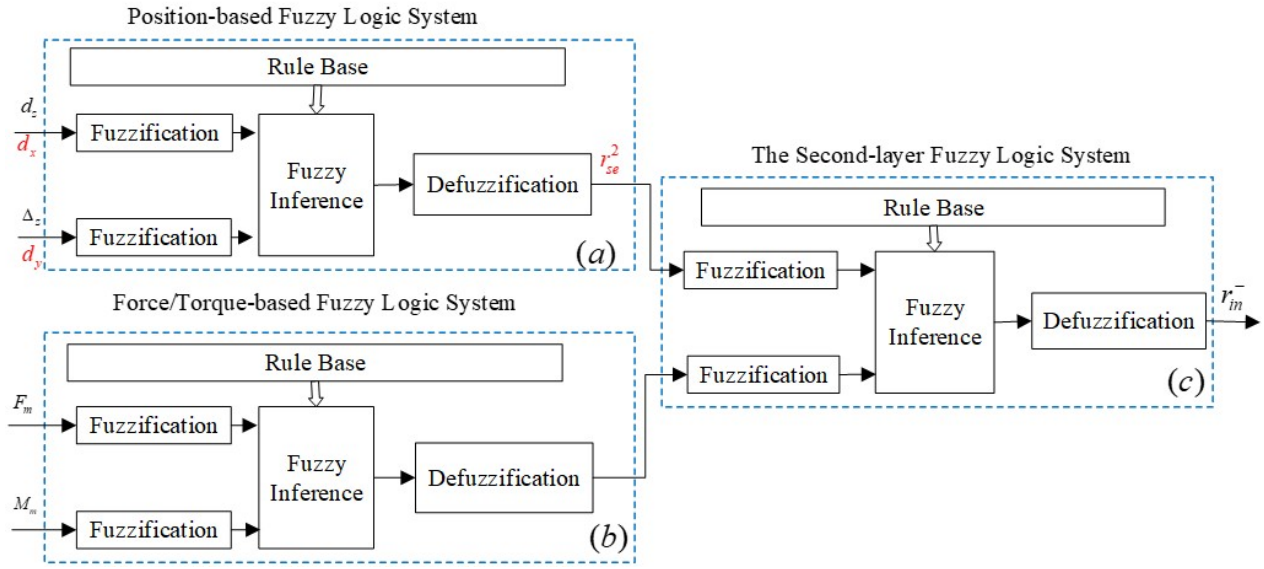

Figure 3. Fuzzy reward system.

The critical reward r_{se}^3 is set as a sparse reward:

$$r_{se}^3 = \begin{cases} -2, & \text{search failure} \\ 1, & \text{search success} \\ 0, & \text{else} \end{cases}. \tag{10}$$

3.2.3. Fuzzy quality evaluation in insertion stage

The reward in the insertion stage consists of two parts, positive reward r_{in}^+ and negative reward r_{in}^-:

$$r_{in} = \omega_{in}^+ r_{in}^+ + \omega_{in}^- r_{in}^-, \tag{11}$$

where r_{in} is the final reward, with ω_{in}^+ and ω_{in}^- representing the weights of each part, with values of 14, 8, respectively. The negative reward is related to the number of assembly steps k_{in} in the insertion stage:

$$r_{in}^- = -\frac{k_{in}}{k_{in}^m}, \tag{12}$$

where k_{in}^m represents the maximum number of assembly steps allowed in the insertion stage to limit the maximum assembly time. r_{in}^- ranges within $[-1,0]$. After preprocessing the required parameters, they are input into the fuzzy control system. The triangular membership function in Equation (8) is used to fuzzify the input parameters into five fuzzy sets. After fuzzifying each parameter, the operation is performed based on the established fuzzy rules and fuzzy inference. The fuzzy rule bases for each layer are established as shown in Figure 4.

Finally, based on the fuzzy set obtained from fuzzy inference, the defuzzification operation is performed using the centroid method to obtain the final positive reward value. The positive reward r_{in}^+ in the insertion stage is obtained through the fuzzy quality evaluation system, which uses four assembly parameters as input: current assembly depth d_z, assembly action Δ_z, and the maximum contact force/torque Γ_m at time t during the insertion process:

$$d_z = P_z^t - P_z^h, \tag{13}$$

$$\Delta_z = P_z^t - P_z^{t-1}, \tag{14}$$

$$\Gamma_m = \max(|\Gamma_x|, |\Gamma_y|, |\Gamma_z|). \tag{15}$$

where P_z^t and P_z^{t-1} represent the z-axis coordinates of the robot at time t and $t-1$, respectively. P_z^h represents the z-axis coordinate of the robot when it reaches directly above the hole plane. $|\Gamma_x|, |\Gamma_y|, |\Gamma_z|$ represent the forces/torques in the x, y and z directions, respectively. The four parameters are processed using a two-layer fuzzy logic structure. The first layer consists of two fuzzy logic systems: the position fuzzy logic system takes the current assembly depth d_z and the assembly action Δ_z as inputs, and the force/torque fuzzy logic system uses the maximum contact force/torque Γ_m at time t as input. The outputs of these two systems serve as the inputs to the second-layer fuzzy logic system, which finally outputs the required reward value. The reward value ranges within [0,1]. The specific structure is shown in Figure 3. The fuzzification and defuzzification methods are the same as those used in the search stage, with the specific values depending on the parameters of the insertion stage.

d_z-Δ_z fuzzy rule base

Δ_z \ d_z	VG	G	M	B	VB
VG	1.0	0.9	0.8	0.6	0.5
G	1.0	0.9	0.7	0.5	0.4
M	0.9	0.8	0.7	0.5	0.3
B	0.8	0.6	0.4	0.2	0.1
VB	0.7	0.5	0.4	0.3	0.1

Force-moment fuzzy rule base

Torch \ Force	VG	G	M	B	VB
VG	1.0	1	0.9	0.8	0.7
G	1.0	0.9	0.8	0.7	0.6
M	0.8	0.7	0.6	0.4	0.4
B	0.6	0.5	0.4	0.3	0.2
VB	0.4	0.4	0.3	0.2	0.1

The fuzzy rule base of the second layer

	VG	G	M	B	VB
VG	0.11	0.16	0.23	0.33	0.48
G	0.11	0.16	0.33	0.33	0.48
M	0.16	0.23	0.33	0.48	0.69
B	0.16	0.23	0.48	0.69	1.00
VB	0.23	0.33	0.48	0.69	1.00

Figure 4. The fuzzy rule base of each layer.

3.3. Staged strategy model for robot peg-in-hole assembly

Both the search and insertion stages of peg-in-hole assembly can be regarded as Markov decision processes, allowing the strategy models for both stages to be acquired through deep reinforcement learning. PPO algorithm is a policy gradient method based on the Actor-Critic framework [24]. As shown in Figure 5, it consists of three networks: a new Actor network, an old Actor network, and a Critic network. The old Actor network's weights are set as the weights of the new Actor network before it is updated.

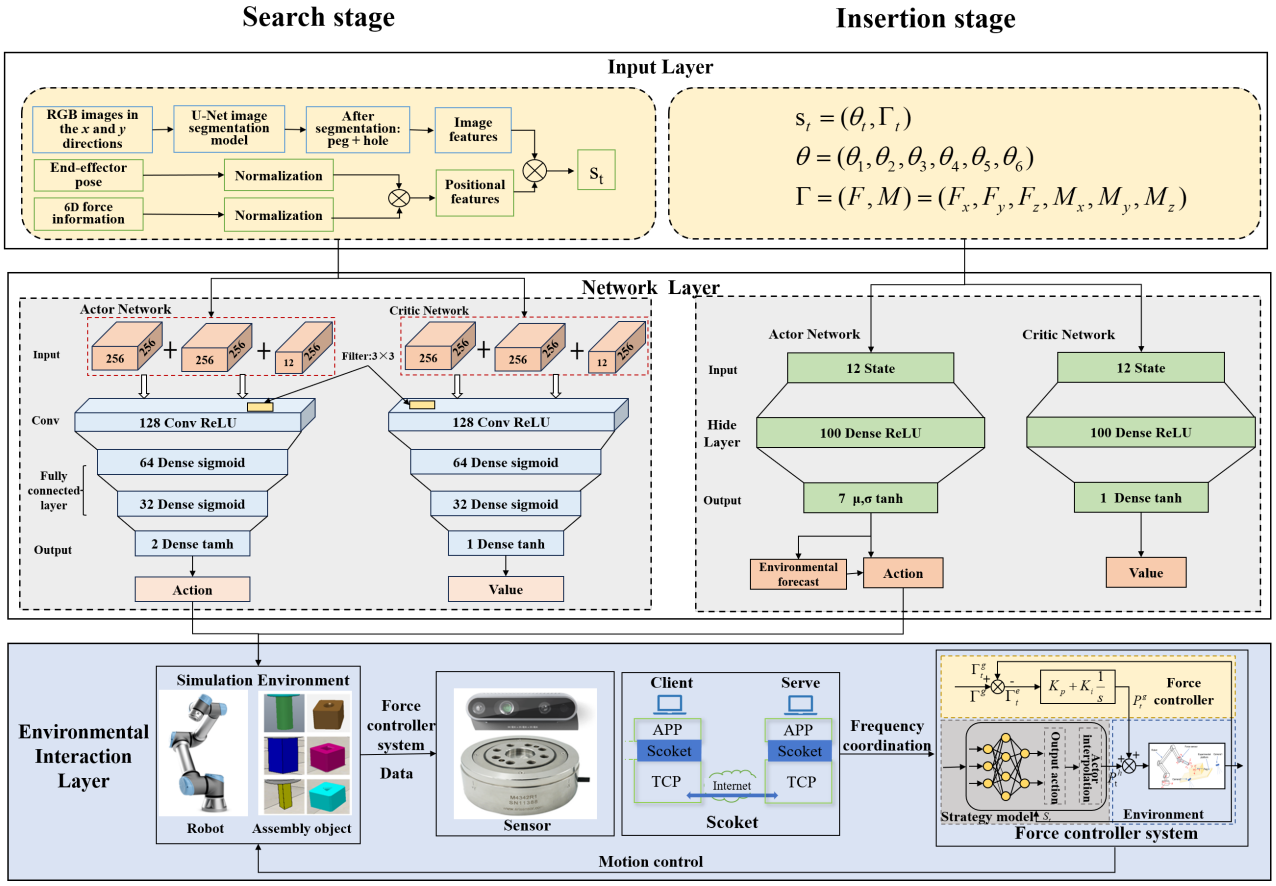

Figure 5. Dual-stage PPO framework for robotic peg-in-hole assembly.

The new and old Actor networks output two normal distributions, $O_\pi(a_t \mid s_t)$ and $O_{\pi_{old}}(a_t \mid s_t)$, based on the current state s_t. Assembly action a_t is sampled from the normal distribution generated by the new Actor network. During training, the data generated by the model's interaction with the environment is temporarily stored in a batch list. The range of stored data is the data generated between the model and the environment after the i interaction and before the $i+l$ interaction. When the model requires updating, the stored data is combined to produce a policy trajectory τ:

$$\tau = (s_{i+1}, a_{i+1}, r_{i+1}, \cdots, s_{i+l}, a_{i+l}, r_{i+l}). \tag{16}$$

where l is the number of times the model interacts with the environment between updates. The Critic network evaluates the current model based on the generated policy trajectory and generates the advantage function $\hat{A}_t$:

$$V'_\varphi = r'_\varphi + \beta V'_\varphi (\varphi \in (i+1, \cdots, i+l)). \tag{17}$$

$$\hat{A}_t = -V_{s_t} + \sum_j V_j, \tag{18}$$

where V_φ is the evaluation of the model by the Critic network at time φ. β is the decay factor, used to reduce the influence of the evaluation from the previous time step on the current evaluation. Based on the advantage function and the normal distributions output by the new and old Actor networks, a truncation method is used to control the Actor network's update gradient:

$$\hbar(\pi) = \mathbb{E}_t[\min(B_t(\pi)\hat{A}_t, \mathrm{clip}(B_t(\pi), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)], \tag{19}$$

where $B_t(\pi)$ is the importance sampling weight, the clip function constrains $B_t(\pi)$ within the range $[1-\varepsilon, 1+\varepsilon]$. ε is a hyperparameter used to control the constraint range of $B_t(\pi)$, which is expressed as:

$$B_t(\pi) = \frac{O_\pi(a_t \mid s_t)}{O_{\pi_{old}}(a_t \mid s_t)}. \tag{20}$$

The algorithm for obtaining the staged strategy for robotic peg-in-hole assembly is based on the PPO algorithm and the fuzzy quality evaluation system, combined with force control to achieve compliant assembly, as shown in Algorithm 1. The search and insertion stages are trained separately, with different reward functions set for each. Two models are ultimately trained, so the reward functions were phase-specific and predetermined before training, with no need for switching during the training process.

Algorithm 1 Robot compliant assembly algorithm based on PPO.

Input: Current state s_t
Output: Assembly action a_t

Initialize the network model
for Maximum assembly times **do**
  for Maximum assembly steps k_{in}^m **do**
    Get the current state s_t
    Obtain the assembly policy based on the current state $\pi(a_t \mid s_t)$
    Get the assembly action a_t using the assembly policy
    Execute the next action a_t and reach the next state s_{t+1}
    Save the interactive datas (s_t, a_t, r_t) to $Batch$
    if size($Batch$) == l **then**
      Get the interaction strategy trajectory $\tau = (s_{i+1}, a_{i+1}, r_{i+1}, \cdots, s_{i+l}, a_{i+l}, r_{i+l})$
      Update the Critic network based on advantage function $\hat{A}_t = -V_{s_t} + \sum_j V_j$
      Update the Actor network based on object function $\hbar(\pi) = \mathbb{E}_t[\min(B_t(\pi)\hat{A}_t, \mathrm{clip}(B_t(\pi), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$
    end if
    if Assembly over (success/failure) **then**
      Turn off force control and return to initial position
    end if
  end for
end for

4. Experiments

4.1. Task description

In this paper, the peg-in-hole assembly is divided into search stage and insertion stage, as shown in Figure 6, both of which can be regarded as Markov decision process. The peg-in-hole assembly is realized through binding force control and contact state recognition. In the search stage, the spatial roaming stage is mainly for the initial estimation of the pose of the hole, which can be estimated by the global camera positioning or force analysis. However, due to problems such as positioning accuracy and occlusion, there will still be a large pose deviation, so it is necessary to further search holes by presetting search trajectory, demonstration learning and strategy learning. In the insertion stage, due to the pose adjustment

error caused by search stage, the inherent accuracy limitation of the robot, sensor error and environmental noise, there are still certain pose deviations between the pegs and holes. Therefore, the jamming and wedging will result in large contact force, so it requires the participation of the force control and the assembly strategy. The force control is to protect the workpieces, prevent the production of large contact force, and speed up the assembly speed, while the assembly strategy is to adjust the robot's pose so that the assembly can be carried out smoothly.

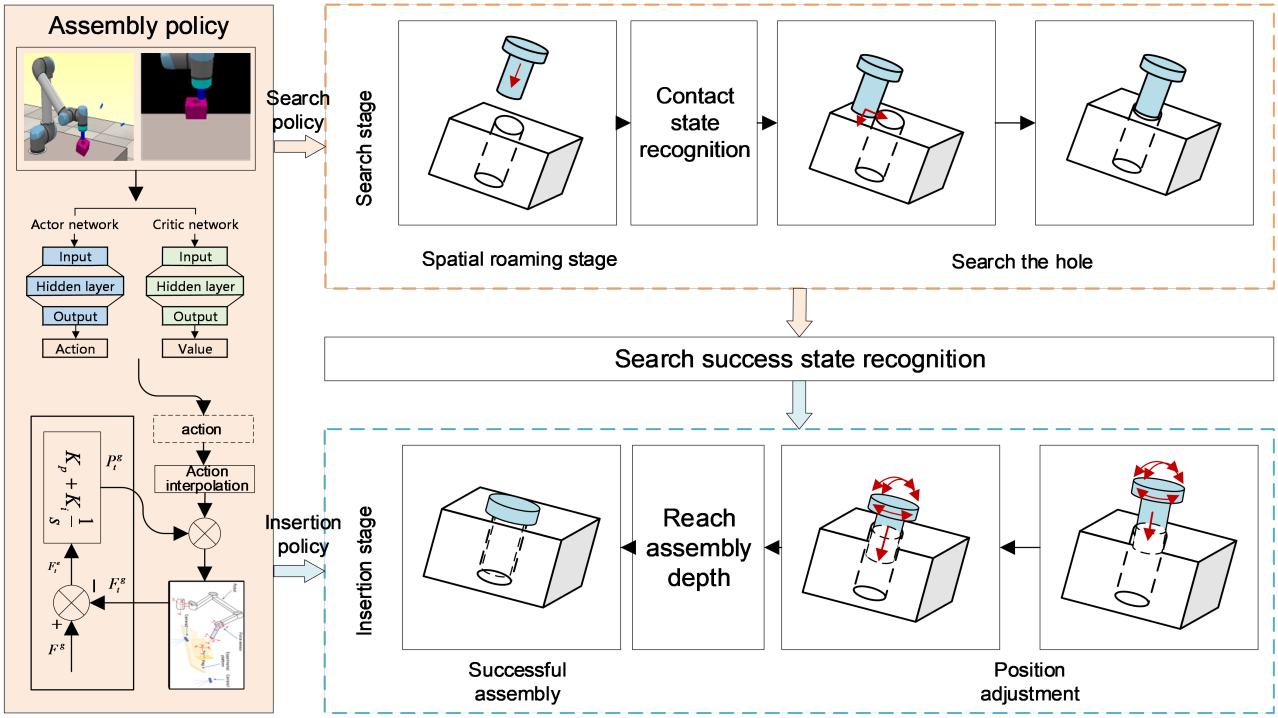

Figure 6. Schematic diagram of the overall process of peg-in-hole assembly.

In order to describe each assembly stage more intuitively, the global situation of the peg-in-hole assembly system constructed in this paper is shown in Figure 7. Firstly, in the space roaming stage, we estimate the pose of the hole based on mechanical analysis, so that the motion coordinate system of the peg $\{T\}$ and the hole plane coordinate system $\{H\}$ are basically parallel. In order to further accurate positioning, the translational hole finding stage is regarded as a Markov decision process. Due to the potential occlusion caused by changes in the relative position of the peg and hole, two cameras are used, placed on either side of the assembly platform with a 90° angle in the *xoy* plane. This setup ensures a more comprehensive observation of environmental changes, and then the search strategy model is obtained by interacting with the assembly environment through the PPO algorithm. The search strategy model outputs the search strategy according to the current interaction state, which is processed by the force control to realize search. During the search process, when the contact force and relative position between the peg and hole appear to change by leaps and bounds, the search is regarded as successful. When the search exceeds the preset search range, it is considered an assembly failure.

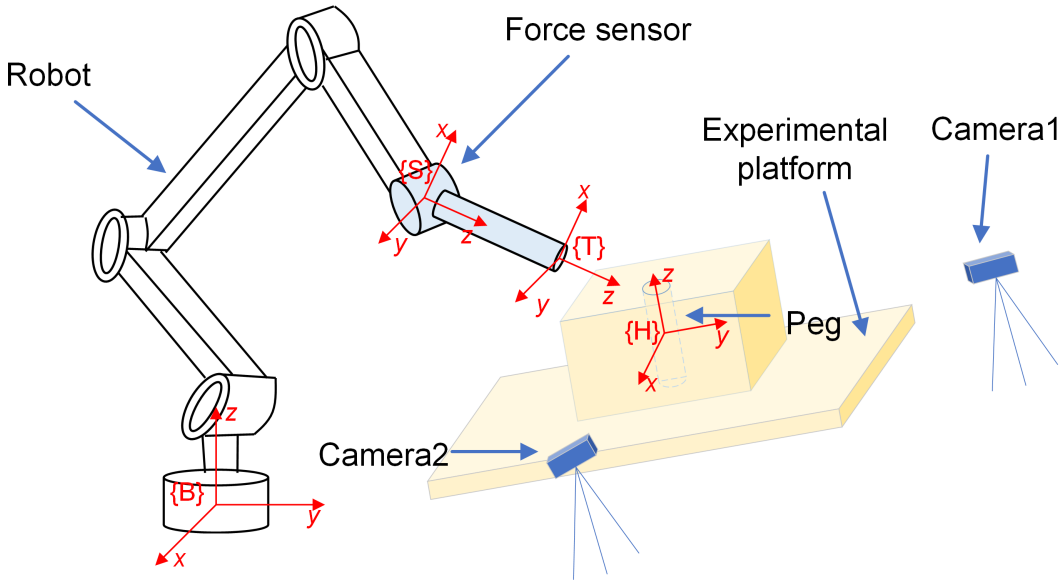

Figure 7. Schematic diagram of the peg-in-hole assembly platform structure.

After detecting the jumps in contact force and relative position, the assembly enters the insertion stage, at which time the relative position between the peg and hole still have some deviation, and the assembly strategy is still needed to adjust the pose of the peg. Similarly, the insertion process is also regarded as a Markov decision process, and the insertion strategy model is obtained through the PPO algorithm, and the insertion strategy achieves the insertion through force control. In the insertion process, when the peg is inserted to a certain depth, the assembly is considered successful, when the contact force and torque exceed the specified maximum range, the assembly is considered failed.

4.2. Experimental platform

The robotic peg-in-hole assembly is designed as shown in Figure 8. The system is divided into three layers: the physical interaction layer, the interaction control layer, and the model strategy layer. The bottom layer is the peg-in-hole assembly platform, which serves as the hardware foundation of the assembly system. The middle layer is the interaction control layer based on the force controller, which forms the core part of the assembly system. The top layer is the strategy algorithm layer used for model learning and training, which is the basis for implementing the assembly strategy. Data and strategy transmission between the layers is achieved through Transmission Control Protocol (TCP) or the Remote Application Programming Interface (API) in the simulation environment.

The peg-in-hole assembly system was built in the simulation and the real world. The simulation environment is based on CoppeliaSim, as shown in Figure 9. The assembly platform is centered around the UR5e collaborative robot, and the key components include the robot, force sensor, visual sensors, assembly platform, data communication system, server, and assembly objects. In this experiment, the force torque (FT) sensor is the one built into the UR5e robot, with a sampling frequency of up to 300 Hz. The camera is the Hikvision MV-CS050-10GC, which can transmit Red, Green, and Blue(RGB) images with a resolution of 2448×2048 pixels, and the peg-hole clearance is 1 mm.

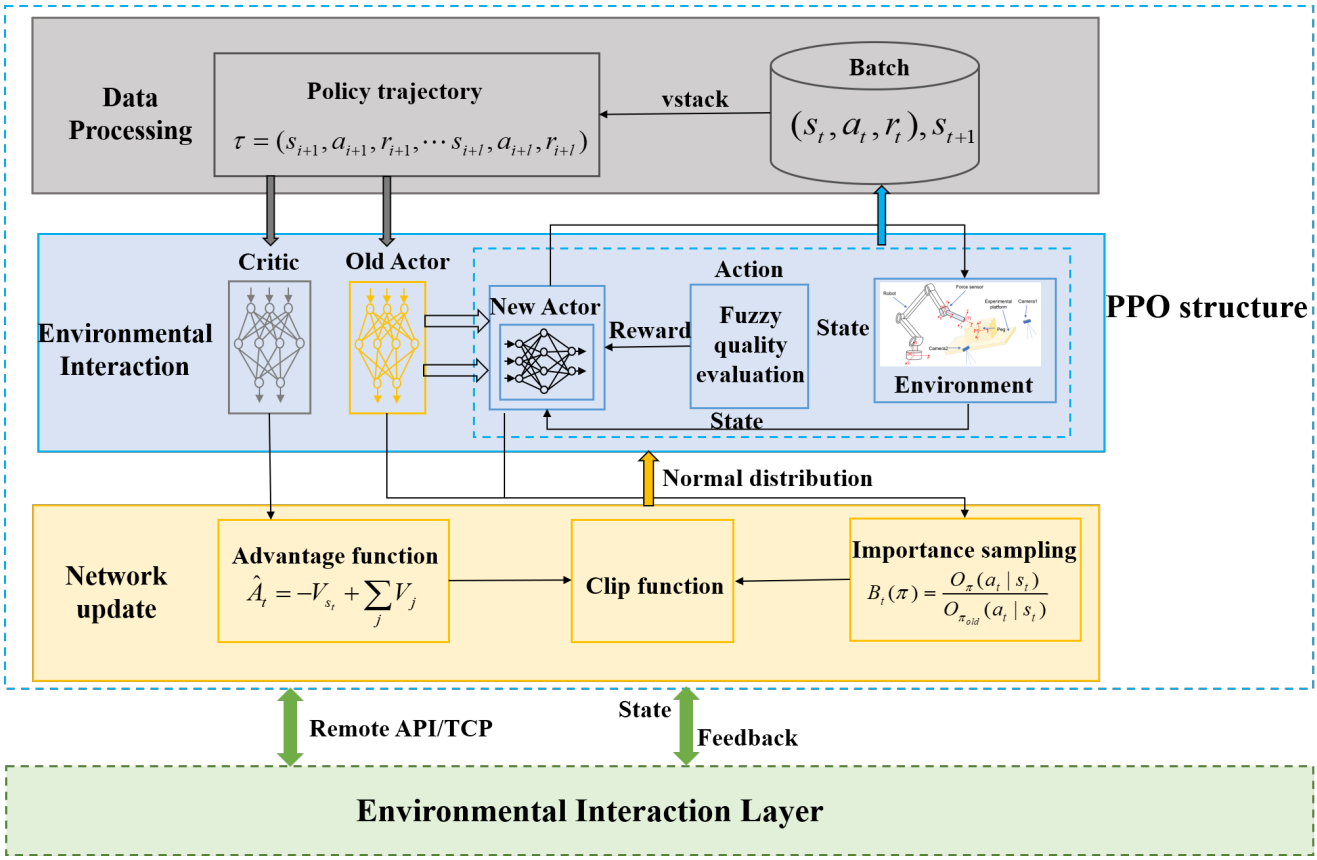

Figure 8. The structure of PPO algorithm.

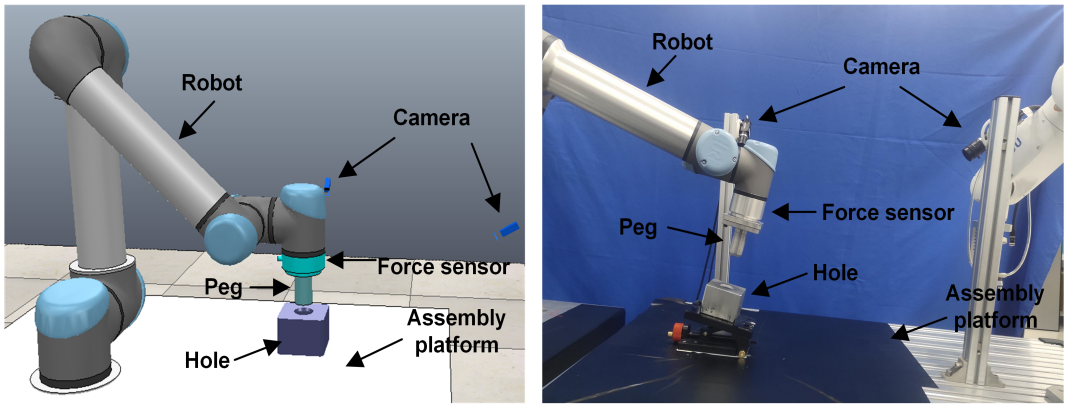

Figure 9. The platform of assembly experiment. **(a)** Simulation; **(b)** Real world.

4.3. Experimental design

Three experiments were designed: validation of the peg-in-hole assembly search strategy model, validation of the peg-in-hole assembly insertion strategy model, and a peg-in-hole assembly experiment with a fixed hole pose. In the insertion experiment, the proposed compliant assembly model was implemented and compared with the DDPG algorithm. To facilitate the study, the following assumptions about the assembly environment were made:

(1) The hole is fixed on the workbench, and the peg is connected to the robot's end-effector via a force sensor, the clearance between peg and hole is 1 mm. Considering that there may still be

pose deviations during the search stage, it is assumed that there is a positional misalignment between the peg and the hole. Before the insertion experiment, set an initial pose deviation of the peg with ΔP_x and ΔP_y range being $(-1, +1)$ mm.

(2) During assembly, force control is applied in all six motion directions of the robot. The frequency of the force controller is 300 Hz, and the policy output frequency is 8 Hz. After frequency coordination, the frequency of both is 200 Hz.

(3) Criteria for determining the end of the assembly: reaching the set depth indicates successful assembly; reaching the maximum number of assembly steps indicates failure. Additionally, a constraint on maximum contact force/torque was imposed during model testing. Trials exceeding the force/torque limit were considered failed attempts.

(4) To verify the robustness and generalization performance of the algorithm, all experiments were conducted in two sets, with different random seeds.

The current policy output frequency is 8 Hz. To align it with the target coordination frequency of 200 Hz, each single action output from the reinforcement learning policy is interpolated into 25 equally spaced sub-actions. This effectively increases the policy output frequency to 8 Hz $\times$ 25 = 200 Hz, which matches the system coordination frequency. The original force controller operates at 300 Hz. To synchronize it to 200 Hz, we perform downsampling on the 300 Hz sampling stream: from every 15 consecutive samples, 10 valid samples are selected. This corresponds to a downsampling ratio of 300/200 = 3/2, which reduces the force control frequency to exactly 200 Hz, ensuring temporal consistency with the upsampled action sequence from the policy. To generate a smooth action trajectory within the policy update interval, a linear interpolation method is adopted, as shown in the following formula:

$$u(t) = a_k + \frac{t}{N}(a_{k+1} - a_k). \tag{21}$$

Here, a_k represents the action vector output by the policy at the k-th step, and a_{k+1} represents the action vector output by the policy at the (k + 1) step. u(t) denotes the interpolated action sent to the robot at the t control step, where N = 25. Using linear interpolation ensures a smooth transition of actions and avoids force control oscillations caused by abrupt action changes.

In the peg-in-hole search stage, the input state of the policy model consists of two components: image features and positional features. The RGB raw images required for image features I are obtained from two visual sensors—one positioned at the front and the other at the side. Image segmentation is employed to isolate the peg and hole, retaining only the relative positional relationship between them. The U-Net segmentation network is used for image segmentation, with training data derived from the image data obtained during the random search process. The images are annotated to build a peg-in-hole image dataset for training, enabling the separation of the peg and hole from the background information.

Positional features consist of the end-effector pose P of the robot and the contact force/torque $\Gamma = (F, M)$, the dimension is 12, which is expanded to 12×256 dimensions through feature encoding. The image features and positional features are vertically concatenated to form the input state (524×256).

$$S = (I, P, \Gamma). \tag{22}$$

The input passes through a convolutional layer to extract state features, followed by two fully

connected layers that transform the 2D features into 1D features. Finally, the output layer generates the final policy. The specific network structure is shown in Figure 10.

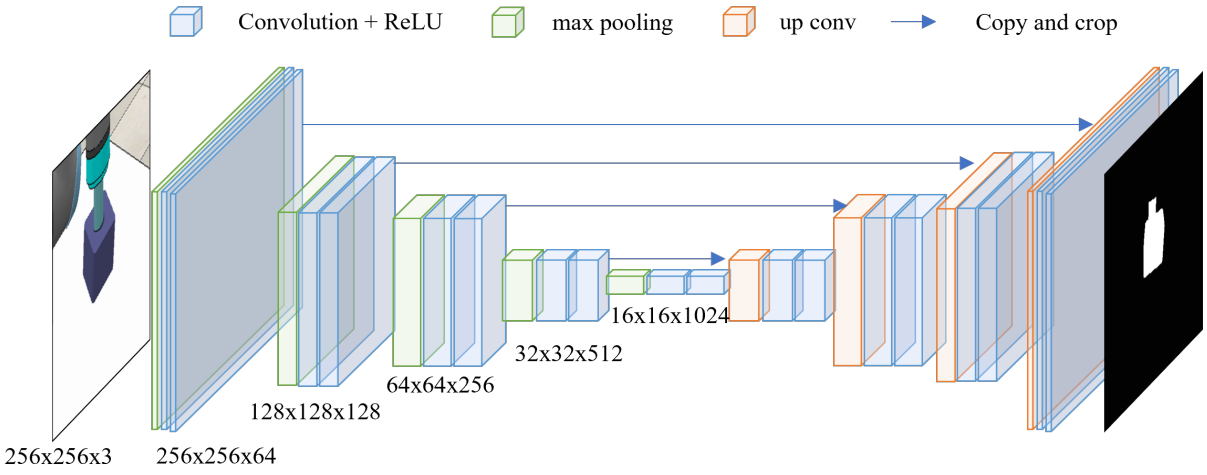

Figure 10. U-Net network architecture.

To verify the reliability of the visual perception module, the U-Net model was used to segment the RGB images captured by the camera and calculate the training metrics. The results show that the U-Net model achieves a segmentation accuracy of 99.46% on the peg-hole segmentation task, with an Intersection over Union (IoU) of 90.63%, Precision of 93.71%, F1 Score of 95.09%, and Dice Coefficient of 95.09%. These excellent quantitative metrics fully demonstrate that the U-Net based visual perception module can accurately extract the relative positional features of the peg and hole, providing stable and reliable visual input for subsequent assembly strategy learning and ensuring the perception accuracy of the entire assembly system.

In the peg-in-hole assembly insertion stage, the input state of the policy model includes the end-effector pose of the robot and the assembly force/torque. The assembly action is represented as a 6-dimensional action vector, used to control the motion of the robot's end effector.

$$s_t = (p_x, p_y, p_z, o_x, o_y, o_z, F_x, F_y, F_z, M_x, M_y, M_z), \tag{23}$$

$$a_t = P_t^h = (\Delta p_x, \Delta p_y, \Delta p_z, \Delta o_x, \Delta o_y, \Delta o_z), \tag{24}$$

$$\begin{cases} k = \frac{b-a}{P_{\max} - P_{\min}} \\ P' = a + k(P - P_{\min}) \end{cases}. \tag{25}$$

where p and o represent the normalized values of the robot's current coordinates and rotation angle, respectively. The normalization formula is shown in Equation (24), where k is the normalization coefficient, $[a, b]$ is the normalization range. P represents the original data, P_{max} and P_{min} are the maximum and minimum values in the data, respectively. P' represents the data after normalization. F and M denote the force and torque generated during assembly; Δp is the translation increment, and Δo is the rotation angle increment.

PI force controller takes the deviation between the desired force F_{des} at the robot end-effector and the actual contact force F_{ext} as input, achieving force closed-loop control through position control. The action a output by the transfer reinforcement learning in this paper includes pose constraints [36]. After superposition,

it serves as the actual assembly action, which is then solved via inverse kinematics and executed by the robot controller. Therefore, the relationship between the PI parameters satisfies $K_P = \rho K_i$ [37].

The maximum number of assembly training iterations u_{max} is set to 1000, with a maximum assembly step count per iteration k_{in}^m is 30; the maximum assembly depth z_{max} is 50 mm; the maximum displacement along the z-axis Δz is 0.8 mm; the maximum force F_{max} is 150 N, and the maximum torque M_{max} is 3 N·m. The desired force $\Gamma^g(F)$ in the x, y, z direction is (0 N, 0 N, 25 N), and the desired torque $\Gamma^g(M)$ is (0 N·m, 0 N·m, 0 N·m). The desired force of 40 N along the z-axis is aimed at enhancing assembly speed.

5. Experimental results and analysis

5.1. Model verification of peg-in-hole assembly search strategy

The results of the peg-in-hole assembly search experiment are shown in Figure 11, which includes the convergence of reward value, search steps, and search time. The horizontal axis represents the number of training iterations. As shown in Figure 11, the reward value of the algorithm increases with the number of training iterations, while the number of assembly steps decreases, indicating that the reinforcement learning-based search strategy model provides effective guidance for the assembly task. The reward value of the final strategy model stabilizes around 19, and both the assembly steps and assembly time converge to approximately 2.5, further demonstrating the superiority of the strategy model.

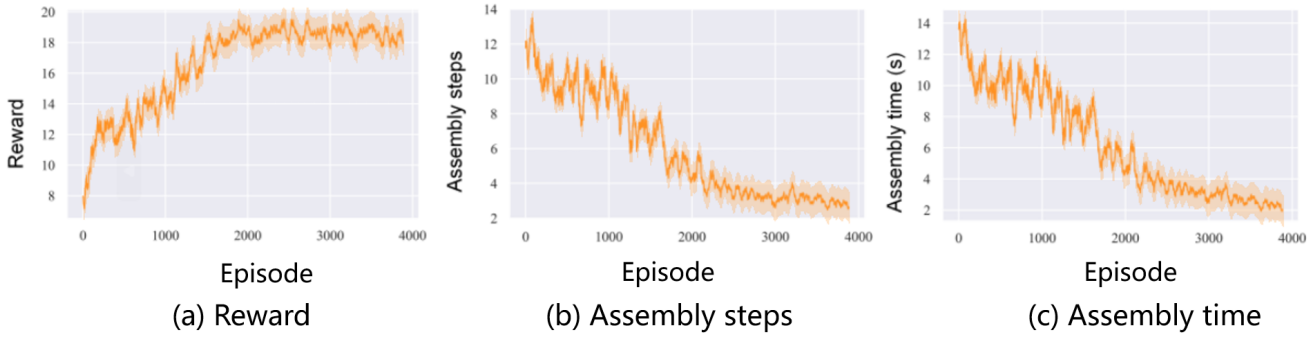

Figure 11. Training result of the search strategy. **(a)** Reward; **(b)** Assembly steps; **(c)** Assembly time.

After training, the acquired strategy model was tested. A total of 5 groups were set up, with 50 assembly tests conducted for each group. The five sets of experiments are identical in terms of model parameters, network structure, and experimental settings, all using the same strategy model. The test success rates are shown in Table 3. The search strategy model achieved a success rate of 100%, demonstrating stable hole-searching performance.

Table 3. Success rate during search stage.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average
Success rate	100%	100%	100%	100%	100%	100%

5.2. *Model verification of peg-in-hole assembly insertion strategy*

Based on the designed fuzzy quality evaluation system, the strategy model was trained in a simulated environment and compared with the DDPG algorithm. The training results are shown in Figure 12, which includes the changes in reward value, assembly steps, and assembly time during the training process, with the horizontal axis representing the number of assemblies. As seen in Figure 12, the reward values of both algorithms continuously increased as the number of assemblies increased, and both the assembly steps and time decreased, indicating that the deep reinforcement learning model based on fuzzy rewards effectively guides the assembly process, successfully learning the assembly strategy.

Although the final reward values of both the PPO and DDPG algorithms stabilized around 66, the convergence speed of the PPO algorithm was significantly faster than that of the DDPG algorithm, indicating that the strategy model obtained by the PPO algorithm is superior to the one obtained by the DDPG algorithm. Moreover, the final assembly steps and assembly time for the PPO algorithm were lower than those of the DDPG algorithm, further demonstrating the advantage of obtaining assembly strategies using the PPO algorithm.

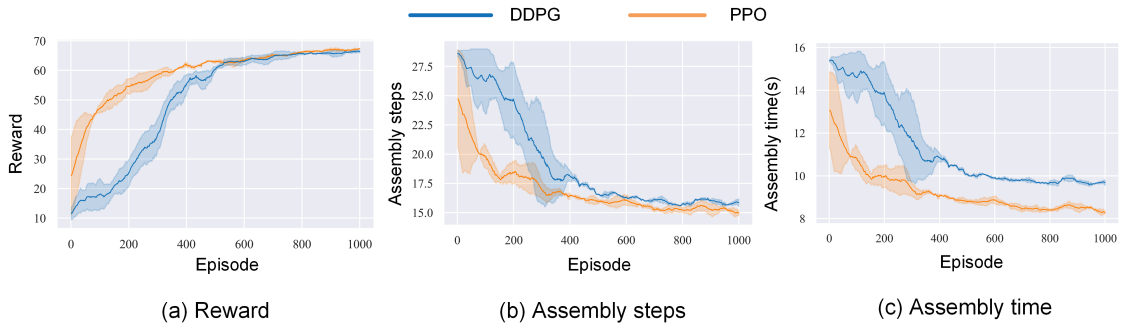

Figure 12. Training result of PPO and DDPG algorithms for assembly strategy. **(a)** Reward; **(b)** Assembly steps; **(c)** Assembly time.

After the training, the strategy models obtained from different algorithms were tested. A total of 5 groups were tested, with each group undergoing 50 assembly tests. The statistical results for assembly steps and assembly force/torque during the tests are shown in Figure 13 and Figure 14 displays the distribution of the robot's end-effector positions during assembly for different algorithms. In the test, the position of the hole is fixed.

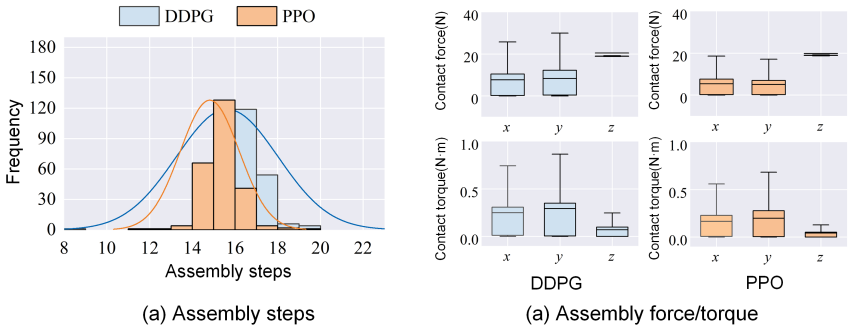

Figure 13. Statistics of assembly steps and force/torque during assembly. **(a)** Assembly steps; **(b)** Assembly force/torque.

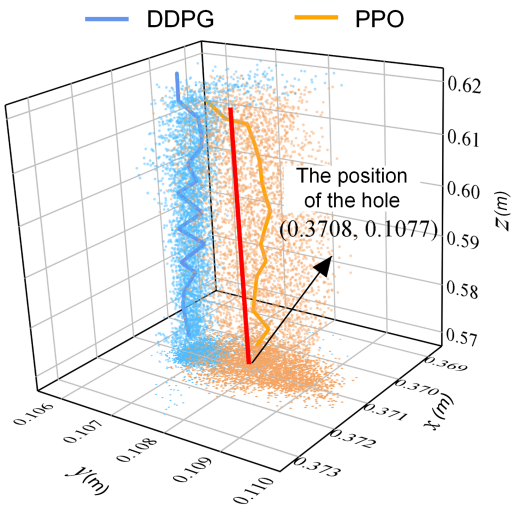

Figure 14. Schematic diagram of insertion trajectory.

In the 250 assembly tests, the assembly steps of the strategy model generated based on PPO were concentrated in 14–16 steps, while those of DDPG were concentrated in 15–18 steps, indicating that the proposed compliant assembly model achieves faster assembly compared to DDPG. Regarding the assembly force/torque during the tests, although the contact forces of both algorithms remained below 40 N and the contact torques within 1 N·m, the maximum contact force of the PPO assembly strategy model was around 20 N, which is better than 35 N achieved by DDPG. This demonstrates that the assembly strategy model designed not only enables compliant assembly but also improves assembly quality by reducing the contact force generated during the assembly process.

From the scatter plot of assembly trajectories in Figure 14, it can be observed that the assembly trajectories based on the PPO algorithm are more dispersed compared to DDPG, indicating that the PPO-based strategy model has explored the environment more thoroughly. In contrast, the DDPG-based search strategy tends to a certain assembly direction, showing insufficient exploration of the environment. Additionally, the PPO assembly trajectories are more closely aligned with the center of the hole.

The test success rates are shown in Table 4, where both PPO and DDPG achieve a 100% assembly success rate. However, PPO maintains a success rate of over 98%, which is 12% higher than 86% achieved by DDPG. Under completely identical experimental settings (same robot platform, task configuration, and training hyperparameters), comparative experiments with the Soft Actor-Critic (SAC) algorithm were supplemented. The results show that the PPO assembly strategy achieves an average success rate of 99.20%, significantly higher than SAC (58%), and the success rate of each individual experiment remains stable above 98%, demonstrating strong robustness. In addition, comparative experiments with TD3 and adaptive impedance control were also included. The average success rate of adaptive impedance control (95%) is slightly lower than that of PPO, and it exhibits significant fluctuations (with a minimum of 89%). This indicates that traditional methods lack sufficient adaptability when dealing with dynamic contact changes during the assembly process. This further validates that PPO, by clipping the objective function to limit the magnitude of policy updates, combined with the designed fuzzy reward system, significantly improves sample efficiency while ensuring training stability.

Table 4. Success rate during insertion stage.

Algorithm	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average
DDPG	96%	100%	94%	94%	86%	94%
PPO	100%	98%	98%	100%	100%	99.2%
SAC	57%	62%	61%	61%	49%	58%
TD3	49%	55%	58%	47%	53%	52.4%
Adaptive Impedance Control	96%	96%	89%	97%	97%	95%

In addition to the comparison with the DDPG algorithm, a comparative experiment was designed to validate the effectiveness of the fuzzy logic reward system. The two sets of experiments differ only in the calculation method of the reward function, with all other settings being the same. Furthermore, to ensure consistency in the scale of the reward functions, the reward range involved in Equations (13)–(15) was normalized to [0,1], consistent with the range of the fuzzy logic reward. In the traditional reward function [38], the weight parameters are set to satisfy $w_1 + w_2 = 1$ and $w_3 + w_4 = 1$, where $w_1 = 0.3$, $w_2 = 0.7$, $w_3 = 0.5$, and $w_4 = 0.5$. The comparison of reward functions during the training process is shown in Figure 15. As shown in the results, the training process with the traditional reward function exhibits slower convergence and lower final reward values. In contrast, the fuzzy logic-based reward function achieves higher convergence values, providing more effective guidance for network training.

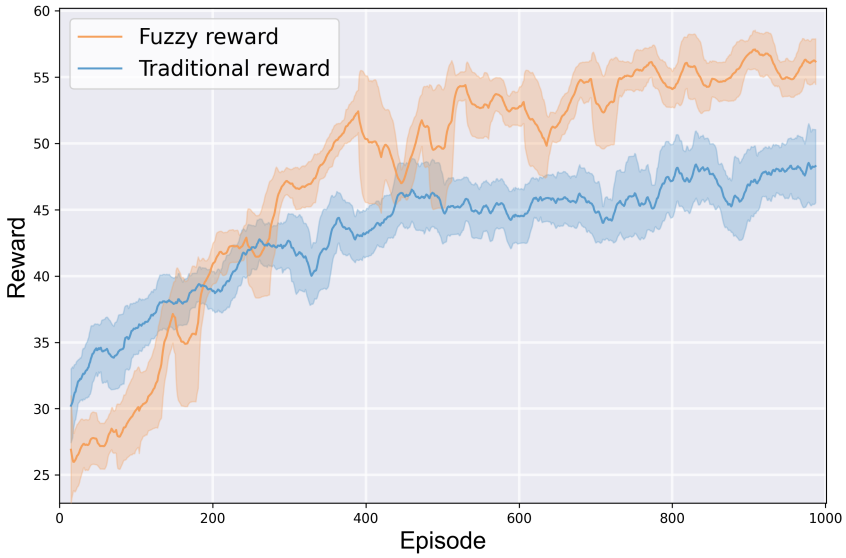

Figure 15. Comparison of training results between fuzzy reward and traditional reward.

To evaluate the impact of the core weight coefficients w_1 and w_2 on assembly performance, multiple sets of comparative experiments were conducted to quantify their effects on assembly success rate, efficiency and stability. This validates the rationality and robustness of the parameter selection in this paper. As shown in Table 5, through four sets of comparative experiments with different weight combinations, all other training conditions were kept identical, and only the values of w_1 and w_2 were adjusted. It can be seen from the table that the combination used in this paper, $w_1 = 0.3$ and $w_2 = 0.7$, performs the best, achieving an average assembly success rate of 99%, with the lowest number of assembly

steps and time. When w_1 is too small (e.g., $w_1 = 0.15$), the proportion of the position reward is insufficient, weakening the algorithm's guidance for peg-hole alignment accuracy, leading to a sharp drop in success rate to 70.80% and an increase in assembly steps. When w_1 is too large (e.g., $w_1 = 0.4$), the algorithm frequently adjusts the pose in pursuit of minimal distance, extending the assembly time to 8.108 s and reducing the success rate to 87.40%. When w_2 is too small (e.g., $w_2 = 0.6$), the force constraint reward is weak and cannot effectively suppress sudden increases in contact force. The success rate drops to 87.40%, posing a risk of force control oscillation. When w_2 is too large (e.g., $w_2 = 0.85$), the assembly time significantly increases (8.178 s), and the success rate decreased to 70.80%.

Table 5. The result of the sensitivity analysis of the weight coefficients.

w	Success rate	Assembly time	Average step
$w_1 = 0.2, w_2 = 0.8$	90.20%	7.628 s	15
$w_1 = 0.15, w_2 = 0.85$	70.80%	8.178 s	16
$w_1 = 0.4, w_2 = 0.6$	87.40%	8.108 s	16
$w_1 = 0.3, w_2 = 0.7$	99.00%	7.318 s	14

5.3. Analysis of the overall process effectiveness in peg-in-hole assembly

Robot carried out the overall process experiment including search and insertion, and the scatter plot of the end-effector trajectory is shown in Figure 16. Throughout the assembly process, the robot fully adapted to the assembly environment, with all trajectory points concentrated within the reachable workspace, successfully completing the peg-in-hole assembly task. This demonstrates that the staged peg-in-hole assembly model effectively explored the environment.

The statistics of the peg-in-hole assembly steps, consisting of both search and insertion steps, is shown in Figure 17. The steps required to complete the assembly task are concentrated between 15–25 steps, indicating that the robot achieved stable and rapid peg-in-hole assembly. This further demonstrates the efficiency and stability of the staged assembly strategy model.

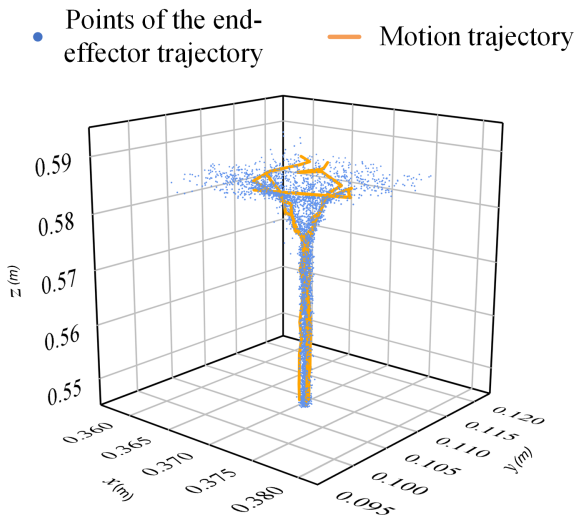

Figure 16. Schematic diagram of overall assembly trajectory in simulation.

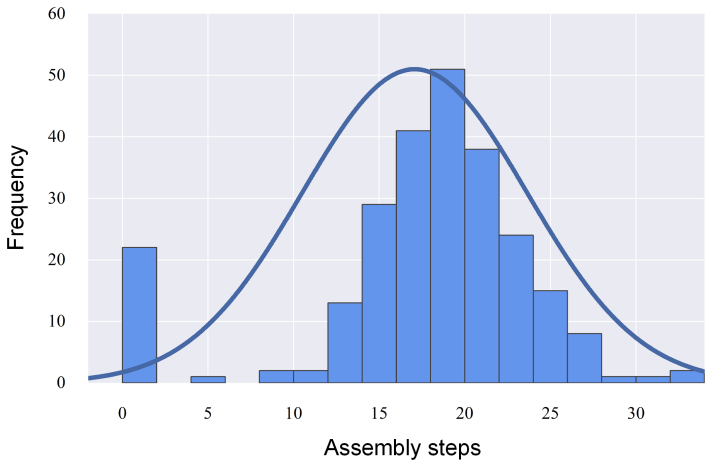

Figure 17. Statistics of assembly steps in simulation.

Figure 18 illustrates the variation curves of force/torque during the assembly process. From the curves, it is evident that the overall process of peg-in-hole assembly exhibits distinct phase characteristics. The contact force remains largely constant during the search stage, while significant forces arise in the insertion stage due to potential jamming and wedging between pegs and holes. However, adjustments in the assembly strategy allow for quick modifications of the robot's pose, effectively preventing excessive contact force/torque and enabling compliant assembly operations. The significant jumps in contact force/torque between the search and insertion stages demonstrate the effectiveness of the contact state recognition algorithm, validating the effectiveness and scientific basis of the robot assembly model based on contact state recognition. The success rate of the robot peg-in-hole assembly is shown in Table 6. The overall assembly success rate reaches 98%, with a 100% success rate in the search stage and an assembly time of only 5.8 seconds. This demonstrates the effectiveness and efficiency of the overall assembly process.

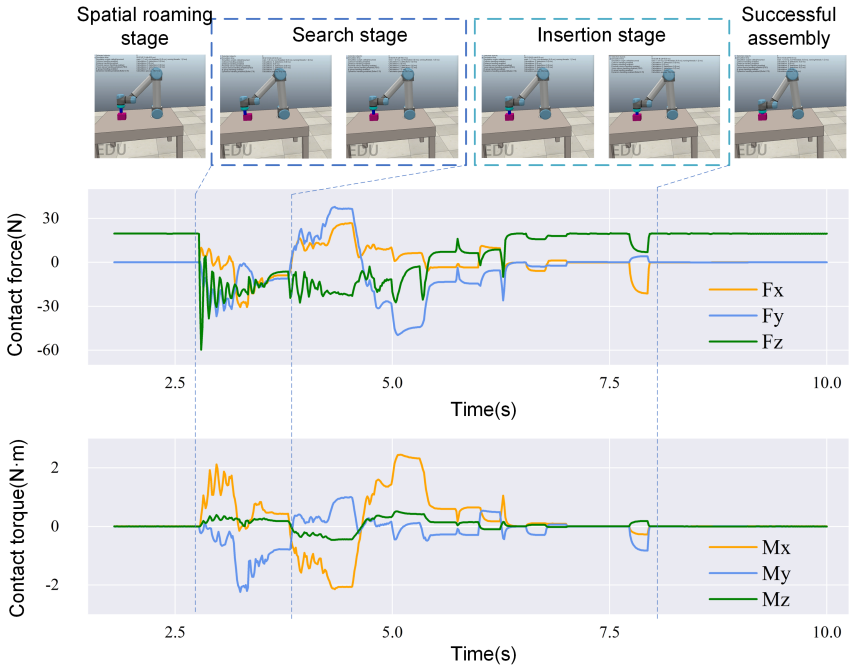

Figure 18. Schematic diagram of phase-wise force/torque variations during assembly.

Table 6. Success rate during insertion stage.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average
Search success rate	100%	100%	100%	100%	100%	100%
Insertion success rate	98%	98%	100%	96%	98%	98%
Assembly time (s)	5.47	5.86	5.96	5.73	5.57	5.72

Through attribution analysis of failure cases in real world experiments, it was found that the main cause of assembly failures in the experiments was hardware constraints. In simulation, the policy model output frequency is 8 Hz and the force controller operates at 300 Hz, which are coordinated to 200 Hz through frequency synchronization. However, in real-world experiments, due to data synchronization issues, discrepancies arise between the policy output and the environment, ultimately leading to assembly failures.

The overall assembly experiment was also validated in the real environment, where the robot's end-effector assembly trajectory and assembly steps were recorded, as shown in Figure 19 and 20. In Figure 19, it can be seen that all trajectory points during the assembly process are within a reasonable range, achieving effective search and insertion. Figure 20 shows the statistics of the total steps during the assembly process, concentrated between 10 and 20 steps, indicating that the proposed method can also achieve efficient and stable assembly in a real environment. We conducted five test experiments. The results can be seen in Table 7. The average success rate is 87%, with the highest reaching 92%. The average finish time is 6.36 s. The training curves can be seen in Figure 21 with initial offset angles of axis at 2°and 5°. 1000 episodes were set for training. It can be seen that the reward value has completely converged at 300 episodes and the number of assembly steps has decreased by 60%.

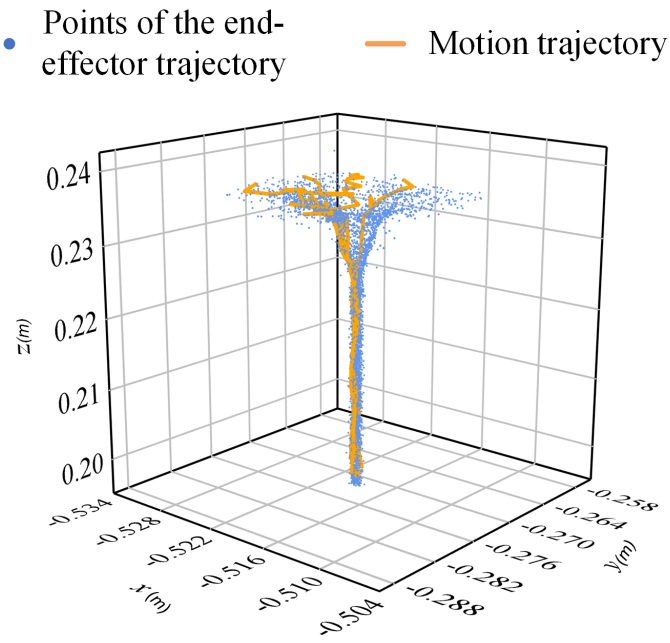

Figure 19. Schematic diagram of overall assembly trajectory in real world.

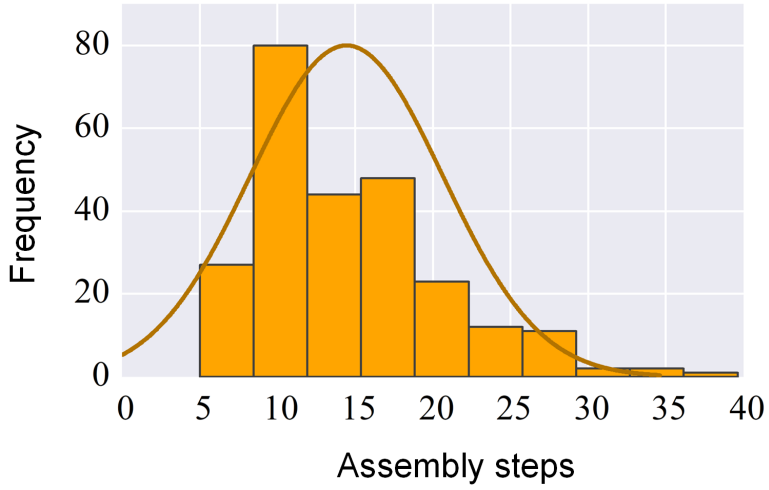

Figure 20. Statistics of assembly steps in real world.

In addition, assembly experiments on a real-world experimental platform involving different materials, shapes, and clearances were added. The experimental results are shown in Table 8. In experiments with different shapes, the assembly success rates for triangular and square peg-hole pairs are comparable, both remaining above 99% under a loose clearance (0.5 mm), demonstrating that the method adapts well to different geometric shapes. Under the 0.5 mm clearance, combinations of different shapes and materials consistently achieve high success rates. Under a small clearance (0.3 mm), the success rate decreases but remains around 80%, reflecting the method's strong generalization ability across different fit clearances.

Table 7. Real experimental results.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average
Real search success rate	98%	100%	96%	100%	98%	98%
Actual assembly success rate	82%	84%	90%	88%	92%	87%
Real average assembly time (s)	6.08	6.11	6.37	6.92	6.31	6.36

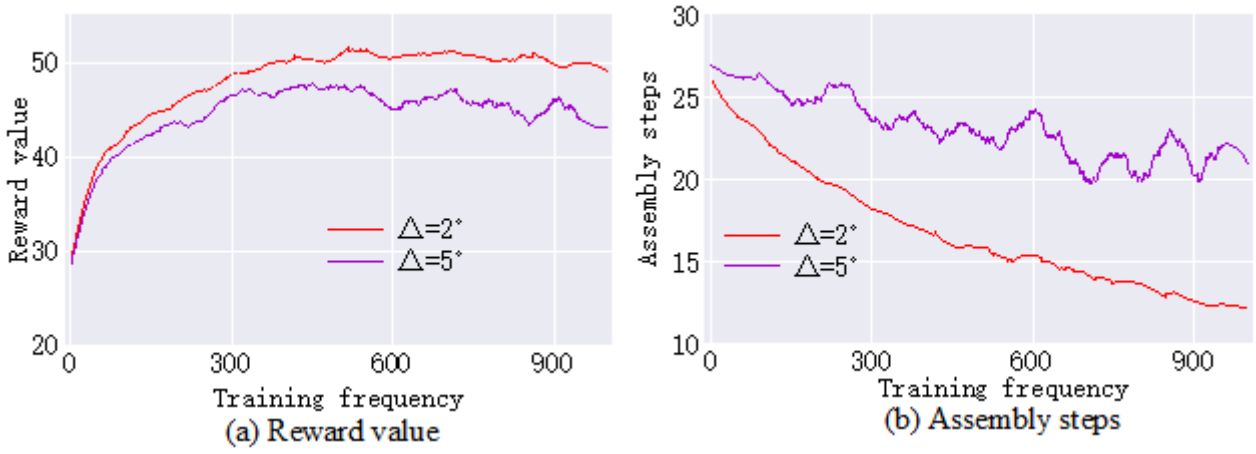

Figure 21. Real experimental training results. **(a)** Reward value; **(b)** Assembly steps.

Table 8. Assembly success rates for diverse peg-in-hole configurations.

Shape	Clearance	Material	Success Rate
Triangle	0.5 mm	Aluminum6061	99.0%
Triangle	0.3 mm	Nylon	80.6%
Square	0.5 mm	Aluminum6061	99.4%
Square	0.3 mm	Nylon	80.0%

To evaluate the industrial applicability of the proposed PPO policy model in the peg-hole assembly task, this paper systematically measures and analyzes the time delay at each stage of the system. During the state perception latency stage, the robot end-effector pose and contact force data are collected from the simulation environment and uniformly mapped to the $[-1, 1]$ range through state normalization. Measured results indicate that this process is extremely fast, with an average time of only 0.003 ms. In the policy output latency stage, the Actor-Critic network based on the PPO algorithm performs forward inference on the normalized state to generate corresponding control actions. Benefiting from the lightweight network design, the average inference time is 32.75 ms. In the force control execution latency stage, the generated control actions are sent to the simulation joint controller to complete the force control closed loop and return the new system state. The average response time for this stage is 65.00 ms. Based on the above real-time performance analysis, it can be concluded that the PPO policy model designed in this paper exhibits stable and controllable time response characteristics throughout the entire process of state perception, policy inference, and force control execution. The total end-to-end system latency is approximately 97.78 ms, which meets the real-time requirements of the robotic peg-hole assembly task.

To verify the generalization ability of the proposed method, assembly experiments with random hole poses were conducted in a simulation environment. The hole pose in the x-direction varies randomly within the range of $(-10, +10°)$. In the search phase, the search strategy model is trained, while the insertion strategy model is obtained through insertion strategy transfer learning using a cylindrical model. Based on the acquired search strategy model and insertion strategy model, assembly skill transfer learning is performed under random hole poses. Due to the randomness of the hole pose, relative pose adjustments based on mechanical analysis are incorporated during the spatial exploration phase. During the assembly process, the maximum contact force is 70 N, and the maximum contact torque is 3.5 N·m, achieving compliant robotic assembly operations under random hole pose conditions. The experimental results are shown in Table 9. In the simulation environment, the overall assembly success rate reaches 96%, with the search phase still achieving a 100% success rate, and the assembly time only requiring 6.496 s.

Table 9. Peg-in-hole assembly experiment under random poses.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average
Search success rate	100%	100%	100%	100%	100%	100%
Insertion success rate	94%	100%	98%	96%	92%	96%
Assembly time (s)	6.30	6.65	6.44	6.64	6.57	6.496

In addition, generalization experiments with different geometric shapes and assembly clearances were conducted. The results are shown in Table 10. The results indicate that the proposed method maintains a high success rate in peg-hole assembly tasks with square, triangular, and other shapes, as well as in circular peg-hole assemblies with different clearances, demonstrating good generalization ability.

Table 10. Experiments with different shapes and clearances.

	Success rate	Average step	Assembly time
Square	98.60%	12.5	7.11 s
Triangle	80.60%	10.0	6.04 s
0.204mm	97.30%	12.1	7.13 s
0.44mm	100%	12.0	6.44 s

5.4. Ablation experiments

Ablation experiments are essential for evaluating the contribution of each core module to overall performance. To this end, three sets of ablation experiments were designed to assess the effects of force-position fusion search, fuzzy rewards, and phased learning, respectively. The details are as follows: In the force-position fusion search experiment, force and position were removed separately and compared with force-position fusion search. In the fuzzy reward experiment, fuzzy rewards were removed and replaced with a traditional deterministic reward function. In the phased learning experiment, test results with and without phased learning were verified separately. All experiments were trained and tested in the same simulation environment, and three metrics—assembly success rate, average number of steps, and average assembly time—were statistically analyzed. The specific experimental results are shown in Table 11.

Table 11. Ablation experiments.

	Success rate	Average step	Assembly time
force-position	100%	6.8	6.7 s
only position	74.40%	9.2	10.2 s
only force	63.70%	12.3	16.6 s
General reward function	86.00%	17.2	7.6 s
Fuzzy rewards	93.80%	14.0	6. 9 s
Phased learning	95.60%	15.5	7.9 s
Without Phased learning	91.30%	17.0	8.3 s

The results indicate that compared to force-position fusion search (100%), the single-modal search approaches—only position (74.40%) and only force (63.70%)—exhibit significantly lower success rates, along with substantial increases in both steps and time. In contrast, the force-position fusion search achieves a 100% assembly success rate, reducing steps and time to 6.8 and 6.7 seconds, respectively. This demonstrates that the fusion of force and position modalities effectively compensates for the limitations of single-modal perception, significantly enhancing both the stability and efficiency of hole localization. When the fuzzy reward function is replaced with a traditional reward function, the success rate drops by 7.8%, assembly steps increase by 3.2, and although assembly time slightly decreases, both efficiency and stability decline noticeably. Phased learning improves the success rate by 4.3%, reduces steps by 1.5, and

shortens assembly time by 0.4 seconds. These experimental results confirm that each core module plays a critical role in assembly performance.

The current work is indeed primarily validated in a simulation environment. However, the peg-in-hole assembly task is highly sensitive to real-world disturbances such as sensor noise, calibration errors, part tolerances, and friction variations. The transfer from simulation to the real environment is key to determining the practical application value of the algorithm. During the simulation design phase, we have already enhanced the generalization ability and robustness of the policy through domain randomization, specifically including:

(1) Applying random perturbations within reasonable ranges to physical parameters such as environmental friction coefficients, end-effector mass, and link inertia;

(2) Introducing pose offsets and contact force fluctuations to simulate calibration errors and part tolerances during the assembly process.

Through the above methods, the generalization ability of the policy under diverse simulation distributions is improved, aiming to narrow the gap between simulation and reality and lay the foundation for subsequent transfer to real robot platforms.

The transfer and generalization of robot assembly skills is a critical issue in current robotics research. This paper addresses the need for transfer and generalization of peg-in-hole assembly skills by revealing the assembly characteristics at different stages through mechanical analysis. The focus is on developing a staged strategy model for robot peg-in-hole assembly and constructing a quality evaluation system. A robot assembly strategy model based on PPO algorithm is proposed, and overall assembly experiments are conducted on both the simulation and real assembly platforms to validate the feasibility of the proposed algorithms.

In the future, we will study the transfer and generalization of assembly strategy between different assembly objects, which is applicable to the peg-in-hole assembly tasks with different clearances, materials and shapes, to improve assembly efficiency.

Data availability statement

The data or datasets that support the findings of this study are available from the corresponding author upon reasonable request.

Declaration of generative AI and AI assisted technologies

During the preparation of the manuscript, the author used generative AI tools (for example, DeepSeek and ChatGPT) solely for language polishing to improve readability, and not for generating the scientific content, methods, experimental results, data analysis, or conclusions of the paper. The authors take full responsibility for the content of the manuscript.

Acknowledgments

This work was supported in part by Grant No. 2024ZY01049, in part by the State Key Laboratory of Robotics under Grant 2023-O02, in part by the General Program of the National Natural Science Foundation of China under Grant 62373225.

Authors' contribution

Li Fengming: writing—original draft preparation, writing—review and editing, data curation, supervision; Qi Hui: writing—original draft preparation, investigation; Jin Ligang: writing—review and editing, software, methodology; Yao Xiaoqing: writing—original draft preparation, validation, visualization; Men Yu: writing—original draft preparation, data curation, formal analysis; Song Rui: writing—review and editing, conceptualization, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare that they have no conflicts of interest.

References

[1] Chen H, Xu J, Zhang B, Fuhlbrigge T. Improved parameter optimization method for complex assembly process in robotic manufacturing. *Ind. Robot* 2017, 44(1):21–27.

[2] Tan Q, Tong Y, Wu S, Li D. Anthropocentric approach for smart assembly: integration and collaboration. *J. Rob.* 2019, 2019(1):3146782.

[3] Edwards C, Edwards A, Stoll B, Lin X, Massey N. Evaluations of an artificial intelligence instructor's voice: social identity theory in human-robot interactions. *Comput. Hum. Behav.* 2019, 90:357–362.

[4] Wu B, Qu D, Xu F. Improving efficiency with orthogonal exploration for online robotic assembly parameter optimization. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Zhuhai, China, December 6–9, 2015, pp. 958–963.

[5] Pfeifer R, Iida F, Bongard J. In *Embodied artificial intelligence: toward building artificial agents that develop adaptive sensorimotor skills*. Cambridge: Elsevier, 2023.

[6] Wu B, Qu D, Xu F. Industrial robot high precision peg-in-hole assembly based on hybrid force/position control. *J. Zhejiang Univ. Eng. Sci.* 2018, 52:379–386.

[7] Xu J, Hou Z, Liu Z, Qiao H. Compare contact model-based control and contact model-free learning: a survey of robotic peg-in-hole assembly strategies. *arXiv* 2019, arXiv:1904.05240.

[8] Xu J, Liu K, Pei Y, Yang C, Cheng Y, *et al.* A noncontact control strategy for circular peg-in-hole assembly guided by the 6-dof robot based on hybrid vision. *IEEE Trans. Instrum. Meas.* 2022, 71:1–15.

[9] Kai W, Rongkang C, Qi C, Weihua L. Robotic assembly of deformable linear objects via curriculum reinforcement learning. *IEEE Rob. Autom. Lett.* 2025, 10:4770–4777.

[10] Zhang K, Wang C, Chen H, Pan J, Wang MY, *et al.* Vision-based six-dimensional peg-in-hole for practical connector insertion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, May 29–June 2, 2023, pp. 1771–1777.

[11] Li X, Chen W, Duan J. Tactile-visual fusion for embodied robotic assembly: a deep reinforcement learning approach. *IEEE Trans. Ind. Inf.* 2025, 11(3):1876–1885.

[12] Kuk-Hyun A, Minwoo N, Jae-Bok S. Robotic assembly strategy via reinforcement learning based on force and visual information. *Rob. Auton. Syst.* 2023, 164:104399.

[13] Van Wyk K, Culleton M, Falco J, Kelly K. Comparative peg-in-hole testing of a force-based manipulation controlled robotic hand. *IEEE Trans. Rob.* 2018, 34(2):542–549.

[14] Park H, Park J, Lee DH, Park JH, Bae JH. Compliant peg-in-hole assembly using partial spiral force trajectory with tilted peg posture. *IEEE Rob. Autom. Lett.* 2020, 5(3):4447–4454.

[15] Lee H, Park J. Contact states estimation algorithm using fuzzy logic in peg-in-hole assembly. In *2020 17th International Conference on Ubiquitous Robots (UR)*, Kyoto, Japan, June 22–26, 2020, pp. 355–361.

[16] Zhang K, Xu J, Chen H, Zhao J, Chen K. Jamming analysis and force control for flexible dual peg-in-hole assembly. *IEEE Trans. Ind. Electron.* 2018, 66(3):1930–1939.

[17] Hou Z, Li Z, Hsu C, Zhang K, Xu J. Fuzzy logic-driven variable time-scale prediction-based reinforcement learning for robotic multiple peg-in-hole assembly. *IEEE Trans. Autom. Sci. Eng.* 2020, 19(1):218–229.

[18] Kim S, Lee J, Park H. Embodied intelligence for adaptive robotic assembly in high-Mix low-volume production. *CIRP Ann.-Manuf. Techn.* 2024, 73(1):1–6.

[19] Xin Z, Huan Z, Pengfei C, Han D. Model accelerated reinforcement learning for high precision robotic assembly. *Int. J. Intell. Rob. Appl.* 2020, 4:202–216.

[20] He W, Dong Y. Adaptive fuzzy neural network control for a constrained robot using impedance learning. *IEEE Trans. Neural Networks Learn. Syst.* 2017, 29(4):1174–1186.

[21] Yang Q, Dürr A, Topp EA, Stork JA, Stoyanov T. Variable impedance skill learning for contact-rich manipulation. *IEEE Rob. Autom. Lett.* 2022, 7(3):8391–8398.

[22] Ning R, Liu Y, We J, Yu S. Posture estimation and adjustment for robotic precise peg-in-hole based on fuzzy variable admittance control. In *2024 14th Asian Control Conference (ASCC)*, Dalian, China, July 5–8, 2024, pp. 2237–2242.

[23] Zhang K, Shi M, Xu J, Liu F, Chen K. Force control for a rigid dual peg-in-hole assembly. *Assem. Autom.* 2017, 37(2):200–207.

[24] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. *arXiv* 2017, arXiv:1707.06347.

[25] Lillicrap T. Continuous control with deep reinforcement learning. *arXiv* 2015, arXiv:1509.02971.

[26] Peters J, Schaal S. Natural actor-critic. *Neurocomputing* 2008, 71(7–9):1180–1190.

[27] Gu S, Lillicrap T, Sutskever I, Levine S. Continuous deep q-learning with model-based acceleration. In *International conference on machine learning*, New York, USA, June 20–22, 2016, pp. 2829–2838.

[28] Gu S, Holly E, Lillicrap T, Levine S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, Singapore, May 29–June 3, 2017, pp. 3389–3396.

[29] Zou P, Zhu Q, Wu J, Xiong R. Learning-based optimization algorithms combining force control strategies for peg-in-hole assembly. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA, October 25–29, 2020, pp. 7403–7410.

[30] Ma X, Xu D. Automated robotic assembly of shaft sleeve based on reinforcement learning. *Int. J. Adv. Manuf. Technol.* 2024, 132(3):1453–1463.

[31] Lee MA, Zhu Y, Zachares P, Tan M, Srinivasan K, *et al.* Making sense of vision and touch: learning multimodal representations for contact-rich tasks. *IEEE Trans. Rob.* 2020, 36(3):582–596.

[32] Schoettler G, Nair A, Luo J, Bahl S, Ojea JA, *et al.* Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA, October 25–29, 2020, pp. 5548–5555.

[33] Hou Z, Fei J, Deng Y, Xu J. Data-efficient hierarchical reinforcement learning for robotic assembly control applications. *IEEE Trans. Ind. Electron.* 2020, 68(11):11565–11575.

[34] Liu Q, Ji Z, Xu W, Liu Z, Yao B, *et al.* Knowledge-guided robot learning on compliance control for robotic assembly task with predictive model. *Expert Syst. Appl.* 2023, 234:121037.

[35] Yan S, Tao X, Ma X, Hao T, Xu D. Adaptive meta policy learning with virtual model for multi-category peg-in-hole assembly skills. *IEEE Trans. Ind. Inf.* 2024.

[36] Jin L, Men Y, Song R, Li F, Li Y, *et al.* Robot skill generalization: Feature-selected adaptation transfer for peg-in-hole assembly. *IEEE Trans. Ind. Electron.* 2024, 71(3):2748–2757.

[37] Beltran-Hernandez CC, Petit D, Ramirez-Alpizar IG, Nishi T, Kikuchi S, *et al.* Learning force control for contact-rich manipulation tasks with rigid position-controlled robots. *IEEE Rob. Autom. Lett.* 2020, 5(4):5709–5716.

[38] Men Y, Jin L, Cui T, Li F, *et al.* Policy fusion transfer: The knowledge transfer for different robot peg-in-hole insertion assemblies. *IEEE Trans. Instrum. Meas.* 2023, 72:3528510.