machines

MDPI

Article

Visual–Tactile Fusion and SAC-Based Learning for Robot Peg-in-Hole Assembly in Uncertain Environments

Jiaxian Tang [1], Xiaogang Yuan [2,*] and Shaodong Li [1]

[1] Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, Guangxi University, Nanning 530004, China; 2312391065@alu.gxu.edu.cn (J.T.); lishaodong@gxu.edu.cn (S.L.)
[2] School of Automation, Southeast University, Nanjing 210096, China
* Correspondence: 230248976@seu.edu.cn

Abstract

Robotic assembly, particularly peg-in-hole tasks, presents significant challenges in uncertain environments where pose deviations, varying peg shapes, and environmental noise can undermine performance. To address these issues, this paper proposes a novel approach combining visual–tactile fusion with reinforcement learning. By integrating multimodal data (RGB image, depth map, tactile force information, and robot body pose data) via a fusion network based on the autoencoder, we provide the robot with a more comprehensive perception of its environment. Furthermore, we enhance the robot's assembly skill ability by using the Soft Actor–Critic (SAC) reinforcement learning algorithm, which allows the robot to adapt its actions to dynamic environments. We evaluate our method through experiments, which showed clear improvements in three key aspects: higher assembly success rates, reduced task completion times, and better generalization across diverse peg shapes and environmental conditions. The results suggest that the combination of visual and tactile feedback with SAC-based learning provides a viable and robust solution for robotic assembly in uncertain environments, paving the way for scalable and adaptable industrial robots.

Keywords: multimodal fusion; robot assembly; peg-in-hole

check for updates

Academic Editor: Dan Zhang

Received: 5 June 2025
Revised: 11 July 2025
Accepted: 12 July 2025
Published: 14 July 2025

Citation: Tang, J.; Yuan, X.; Li, S. Visual–Tactile Fusion and SAC-Based Learning for Robot Peg-in-Hole Assembly in Uncertain Environments. *Machines* **2025**, *13*, 605. https://doi.org/10.3390/machines13070605

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Robotic assembly is a critical and integral part of modern industrial production, playing a significant role in improving efficiency and ensuring product quality. In structured environments, robotic assembly techniques have advanced to the point where they can handle repetitive, high-precision tasks, replacing human labor and improving consistency [1]. However, when assembly tasks occur in unknown or dynamically changing environments, traditional methods relying on exact pose control face significant challenges [2]. Specifically, in tasks such as peg-in-hole assembly, the uncertainty associated with pose deviations, environmental noise, and variations in object shape pose substantial barriers to achieving high-performance robotic assembly in real-world applications [3].

One of the fundamental challenges in robotic peg-in-hole assembly tasks is the inherent uncertainty present in the task's environment [4]. In practical industrial settings, objects to be assembled often vary in shape, size, and material properties, and the presence of environmental noise further exacerbates the complexity. The conventional approach to these assembly tasks often relies on precise motion control [5], assuming a known object pose and a structured environment. However, when facing pose deviations or complex

geometries, these traditional methods are no longer effective [6]. As such, there is a growing need for more robust, adaptive robotic assembly systems capable of handling a wide variety of assembly tasks, particularly in uncertain and dynamic environments.

A promising direction to address these challenges is the integration of multiple sensor modalities [7], such as visual and tactile data, to enhance a robot's environmental perception and facilitate more informed decision-making during assembly tasks. Vision-based systems, commonly used in robotic assembly, can provide high-resolution spatial information about the assembly environment [8]. However, visual systems alone often struggle with tasks requiring precise force control, especially when the robot is engaging with objects in constrained or dynamic environments. Tactile sensors, on the other hand, offer valuable feedback about contact forces and object interactions, providing essential information for tasks that require fine force control, such as peg-in-hole assembly. Combining these complementary sensory inputs—vision for spatial perception and tactile feedback for force control—creates a powerful multimodal system that enhances the robot's overall understanding of the environment [9].

In this paper, we explore the concept of visual–tactile fusion, which involves integrating RGB images, depth images, tactile force information, and pose data into a unified representation. To achieve this, we propose a novel fusion network based on the autoencoder that is capable of processing and combining these multimodal inputs. This approach allows the robot to create a more accurate and comprehensive perception of its environment, significantly improving its ability to deal with uncertainties like pose deviations and environmental changes. The network is designed to extract multimodal features that are then fused to enhance the robot's sensory capabilities, enabling better decision-making during the assembly process. Furthermore, the effectiveness of the multimodal fusion network is validated through extensive simulations, demonstrating its robustness in environments with significant uncertainty.

Building upon this, we introduce the Soft Actor–Critic (SAC) algorithm for assembly skill learning. Reinforcement learning (RL) [10,11], a branch of machine learning inspired by behavioral psychology, enables agents to learn optimal policies by interacting with the environment and receiving feedback in the form of rewards. Unlike supervised learning, which relies on labeled data, RL focuses on trial-and-error experiences, making it particularly well-suited for robotics tasks where labeled trajectories are expensive or infeasible to obtain [12].

Among the various RL algorithms, SAC has emerged as one of the most effective off-policy, model-free algorithms, especially for continuous control problems. SAC combines the benefits of actor–critic architectures with the principle of maximum entropy, encouraging exploration while optimizing performance [13,14]. This leads to improved sample efficiency and robust policy learning in high-dimensional, stochastic environments. In the context of robotic peg-in-hole assembly, the task is characterized by contact-rich dynamics, variable tolerances, and uncertain environmental factors such as deviations in peg position, surface roughness, or unexpected collisions. These conditions pose significant challenges for traditional control methods that rely on pre-defined trajectories or force thresholds. SAC, however, can adapt to these uncertainties by learning policies that continuously refine action selection based on real-time sensory feedback [15].

By integrating SAC with multimodal sensory inputs—including visual information, force–torque signals, and proprioceptive data—our method enables the robot to learn nuanced insertion strategies. The learned policy not only achieves accurate and stable insertions but also demonstrates generalization across different peg shapes and hole configurations. This adaptability is crucial for deployment in real-world industrial scenarios where environmental conditions are rarely ideal or repeatable.

We combine the visual–tactile fusion network with SAC-based learning to create a powerful system for robot assembly. The system learns to adapt its actions based on varying input states that include not only visual and tactile data but also pose information. The SAC-based learning algorithm optimizes the robot's decision-making process, ensuring that it can adapt to a variety of initial conditions, such as different initial positions and peg types. Through experiments, we show that this approach significantly enhances the robot's performance, achieving high success rates across different assembly scenarios and demonstrating improved generalization capabilities when confronted with various peg shapes and assembly tasks [16–20]. This is a key advantage of the SAC algorithm, as it allows for the efficient exploration of different strategies without requiring a large amount of pre-programmed task knowledge [21].

By combining visual and tactile feedback with reinforcement learning, this research aims to advance the field of robotic assembly in uncertain environments. The proposed multimodal fusion network enables robots to better perceive and adapt to complex environments, while the SAC-based learning strategy improves the generalizability of assembly tasks. The results of our experiments demonstrate the effectiveness of this approach in simulation applications, proving that combining vision, tactile feedback, and reinforcement learning provides a viable path toward achieving robust, adaptive, and scalable robotic assembly systems. In this work, we define "uncertain environments" as operational settings that include (1) random initial pose deviations of the peg, (2) variations in peg and hole geometry, (3) force fluctuations due to contact, and (4) noise in visual and tactile sensor readings. These factors collectively reflect the typical variability encountered in real-world robotic assembly tasks and form the core perturbations under which our system is evaluated.

In summary, the primary contributions of this work encompass the following:

- We propose a novel robot assembly framework that combines visual–tactile multimodal perception with reinforcement learning. This fusion enables the robot to perceive environmental uncertainties and object states more comprehensively, which is critical for robust peg-in-hole operations.
- We develop a multimodal feature fusion network based on the convolutional autoencoder; this network can effectively extract and fuse multimodal information (RGB image, depth map, force–torque signals, robot pose information). The fused features provide rich context for decision-making during assembly tasks.
- We integrate the Soft Actor–Critic (SAC) algorithm into the robot control pipeline for adaptive skill learning. By using fused sensory features as input, the SAC-based policy learns to generate precise control actions that are robust to pose deviations and variable contact conditions.

2. Materials and Methods

2.1. Visual–Tactile Fusion Network

Processing multimodal inputs is essential for achieving robust perception in robotic assembly tasks, especially in uncertain or contact-rich environments. Vision-based methods alone often struggle with occlusion, poor lighting, or reflective surfaces, leading to inaccurate pose estimation. Tactile sensing, while precise in capturing contact forces, lacks spatial awareness and cannot predict global object position or orientation. In peg-in-hole tasks, where both force control and precise spatial alignment are required, fusing visual and tactile data allows the robot to compensate for deficiencies in either modality. For instance, visual input helps guide the initial alignment, while tactile feedback assists in detecting contact and guiding insertion even under visual uncertainty. The novelty of our proposed fusion network lies in its autoencoder-based structure, which learns compact and stable

latent representations across RGB, depth, tactile force, and robot pose. Unlike methods that simply concatenate raw features, our design extracts modality-specific encodings before performing learned fusion, enabling better generalization and noise resilience. The first step in improving the robot's performance is to enhance its perception capabilities. In complex assembly tasks, a robot's ability to understand its environment is critical to making effective decisions.

In this study, we define environmental uncertainty in the peg-in-hole assembly process as arising from multiple sources. These include: (1) initial pose deviations of the peg relative to the hole in both position and orientation; (2) variations in contact force dynamics due to surface friction, compliance, or unexpected collisions; (3) sensor noise in visual (RGB and depth) and tactile data streams; and (4) differences in the geometric properties of pegs and holes, such as shape and dimensional tolerance. These uncertainties reflect common real-world scenarios in robotic assembly tasks and pose significant challenges for precise alignment and insertion. Our proposed framework is explicitly designed to perceive and adapt to such uncertainties using multimodal fusion and reinforcement learning.

In this work, we focus on solving the challenges of peg-in-hole assembly in uncertain environments, specifically targeting issues such as pose deviations, environmental noise, and varying peg shapes. The method we propose is based on integrating multimodal perception through visual and tactile feedback and enhancing assembly skill ability via the Soft Actor–Critic (SAC) algorithm. This combination enables the robot to not only perceive its environment more effectively but also learn to perform robust assembly tasks that generalize well across different conditions. Below, we detail the key components of our methodology, including the visual–tactile fusion network, the SAC-based skill learning approach, and the integration of these two elements to achieve robust assembly strategies.

2.2. Visual–Tactile Fusion Network

The first step in improving the robot's performance is to enhance its perception capabilities. In complex assembly tasks, a robot's ability to understand its environment is critical to making effective decisions. In typical robot assembly applications, visual feedback from cameras (RGB and depth images) is often used to detect object positions and orientations. However, in environments where precise force control is required, tactile sensors become indispensable. These tactile sensors measure contact forces between the robot and the object, providing real-time feedback that is crucial for tasks like peg-in-hole assembly, where the robot must adjust its forces to avoid damaging the object or failing the task.

To leverage both visual and tactile data, we design a visual–tactile fusion network based on an autoencoder, which serves as the backbone for processing and combining multimodal data. This network is trained to extract features from RGB images, depth images, tactile force information, and robot pose data, then fuse these features into a unified representation. The key challenges in this fusion process involve aligning the data from these different modalities and dealing with the potentially high-dimensional nature of the combined data. To address this, we use an autoencoder, which is well-suited for dimensionality reduction and feature extraction, making it easier to combine these diverse data sources.

In summary, this paper proposes a visual–touch fusion network based on an autoencoder, mainly consisting of three modules (as shown in Figure 1): feature extraction (the green box part), feature fusion (the blue box part), and decoding prediction (the orange box part). Among them, the purple part is the content of the downstream reinforcement learning strategy.

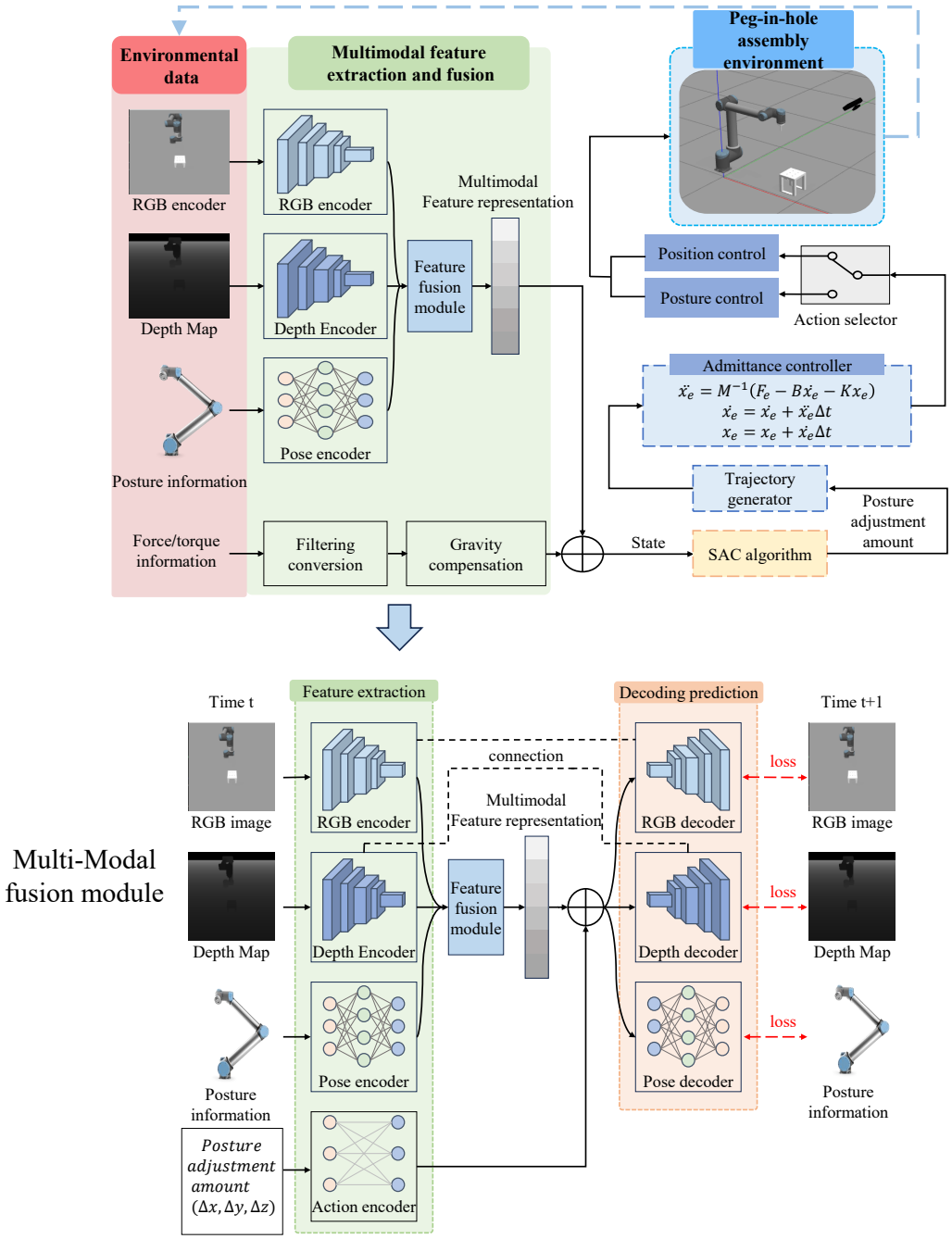


Figure 1. Multimodal assembly skill learning strategy based on SAC algorithm (The left half of the upper part of this figure represents our multimodal fusion section, and the lower part of this figure shows our multimodal fusion module).

2.2.1. Multimodal Feature Extraction

Each sensor modality provides valuable information, but it also comes with its own challenges. For example, RGB image are great for capturing the visual appearance of objects, but they do not provide depth information, which is critical for accurately positioning objects in 3D space. Depth cameras, such as those using stereo vision or time-of-flight sensors, provide accurate spatial information but lack the texture details that RGB cameras can capture. Tactile sensors, on the other hand, provide valuable force feedback, essential for controlling interactions between the robot and the assembly objects, but they cannot directly inform the robot of the object's position or shape.

To handle these challenges, we propose a network architecture that integrates these multimodal features in a way that maintains the integrity of each modality's strengths. Specifically, each sensor modality (RGB, depth, tactile, and pose data) is processed through separate feature extraction modules, such as convolutional layers for visual data and fully connected layers for tactile data. The features from these individual channels are then fused using a multilayer perceptron (MLP) to create a unified feature space. This fusion process allows the robot to have a more comprehensive understanding of its environment by combining visual and tactile information, which improves its ability to perform precise and robust assembly tasks.

In the peg-in-hole hole task, visual information is directly collected by the Kinect V2 camera (Microsoft Corp., Redmond, WA, USA), which is directly fixed in the world coordinate system and is capable of capturing the complete operation space of the assembly task. Visual information refers to RGB image and depth image, and the corresponding raw data are collected by the camera at a rate of 30 frames per second. For the collected raw visual data, image preprocessing operations are carried out first, including operations such as cutting, normalization, smoothing processing, and noise removal to obtain RGB image data of 128 × 128 × 3 and depth image data of 128 × 128 × 1. Then, the preprocessed image data are input into the corresponding neural network for feature extraction. For the input data of RGB images and depth images, this paper adopts a 6-layer convolutional neural network for data encoding and processes the size changes of feature maps through the convolutional layers instead of directly using the pooling layers for processing. Furthermore, in order to further extract potential features, a fully connected layer is added at the end of the feature extraction channels of both RGB images and depth images, converting the encoded feature vectors into 2 × 128-dimensional RGB feature vectors and depth RGB feature vectors. The RGB and depth branches each generate a 2 × 128 feature vector through their respective convolutional encoders and fully connected layers. These vectors remain separate until the fusion stage.

The pose information of the robot is directly read from the robot body, while the pose information of the center at the end of the assembly axis is calculated through the homogeneous transformation matrix, and the pose adjustment amount information is the expected action given by the controller. For the pose information of the robot body, in this paper, a 4-layer Multilayer Perceptron (MLP) is adopted to perform data encoding and feature extraction on the position and pose (the pose is represented by Euler angles) of the center at the end of the assembly axis at the current moment, finally generating a 2 × 128-dimensional pose feature vector. For the pose adjustment information of the robot, this paper adopts a 2-layer MLP to perform data encoding and feature extraction on the current pose adjustment information, finally generating a 32-dimensional action feature vector.

2.2.2. Multimodal Feature Fusion Model

Once the features from the different modalities are extracted, we need to fuse them into a cohesive representation that the robot can use for decision-making. We achieve this by utilizing a multimodal autoencoder. The autoencoder is a deep learning model designed for unsupervised learning, which learns to compress input data into a lower-dimensional latent space and then reconstruct it. This structure allows the autoencoder to capture the essential features of the data while discarding noise.

The neural network method can learn the correlations and representations among different modalities, thereby improving the performance and generalization ability of the model, achieving great success in fields such as images, language, and sound. Therefore, in this paper, the multimodal feature fusion module will be implemented by the neural

network to process and fuse data feature vectors of different modalities. The RGB feature vectors, depth feature vectors, and pose feature vectors output by the feature extraction module are taken as inputs, and the latent representations of the feature data are learned using the feature fusion module, thereby obtaining the final 128-dimensional multimodal feature representation.

In the robot peg-in-hole assembly task for multimodal perception, RGB images, depth images, and robot pose information constitute the input data, while the environmental feedback data at the next moment constitutes the supervision label, and the robot pose adjustment amount constitutes the execution action. The above-mentioned input data, supervision labels, and execution actions together constitute a multimodal dataset $D = (o_i, y_i, a_i)|i = 1, ..., T$. Therefore, in this paper, we adopt the deterministic model fusion method based on the neural network. It uses multilayer neural networks to achieve feature extraction and fusion of different modal information and is capable of learning complex feature representations. In the research work of this paper, the feature extraction module, namely the encoder, has completed the feature extraction of various modal data. Therefore, the feature fusion module does not need to design an overly complex network structure. In this paper, a two-layer multilayer perceptron is adopted as the feature fusion module to complete the feature extraction and fusion of RGB, depth, and pose feature vectors, thereby learning and obtaining a deterministic multimodal feature representation.

Different from the classic autoencoder structure, the decoder designed in this paper does not reconstruct the original input data but accomplishes the prediction task. During the training process of multimodal networks, the 128-dimensional multimodal feature representation and the action feature vector are concatenated, combined, and input into the corresponding decoder to predict the environmental state data at the next moment. The supervised learning objective is continuously optimized by predicting the loss value of the task. The RGB and depth image decoding predictor adopts a 4-layer deconconvolutional neural network and 4 skip connections, it uses the upsampling method to process the action feature vectors, and it finally decodes and predicts the RGB and depth image data at the next moment. The pose decoder is a 4-layer MLP used to predict the pose information at the next moment at the end of the assembly axis. In the deterministic fusion model, the network parameters are updated based on the prediction result loss of the decoder. The endpoint error loss is used for the prediction results of RGB and depth images, and the average processing is performed on all pixels. The mean square error is used as the loss function for the prediction of the end pose information of the assembly axis. After the training of the visual–tactile fusion network based on the autoencoder is completed, the network model is frozen and not updated. The data input from different sensors is fused using the feature extraction module and the feature fusion module to generate a low-dimensional, effective, and stable multimodal feature representation (128-dimensional multimodal feature vector). The combination of this 128-dimensional multimodal feature vector and the contact force information will be used as the environmental state to describe the peg-in-hole assembly process, thereby accelerating the training of the reinforcement learning strategy and improving the stability and generalization of the operation strategy.

To address potential coupling effects between visual and tactile data, our model adopts a modular architecture where each modality is first processed independently. Specifically, RGB, depth, and pose data are passed through separate convolutional or MLP-based encoders, while contact force data is handled separately at the reinforcement learning stage. This separation ensures that low-level feature extraction is modality-specific and not directly influenced by cross-modal noise. Only after independent feature extraction are the encoded representations fused through a learned multilayer perceptron (MLP), allowing the network to learn useful joint representations while avoiding early fusion

conflicts. This design enables the model to preserve modality-specific characteristics and mitigates issues arising from sensor synchronization delays or physical cross-talk between vision and force signals.

2.2.3. Multimodal Data Collection Strategy

In order to obtain more efficient sample data to enhance the performance of the visual–tactile fusion network, this paper designs a reward-based random exploration strategy, which combines the concept of evaluation and reward for agent actions by the environment in reinforcement learning algorithms. This random exploration strategy aims to enable the system to extensively explore the environment or workspace and accumulate a series of high-quality data samples by randomly selecting actions with high reward nature. In this random exploration strategy, it mainly includes three parts: the action generator, the evaluator, and the random number generator. The action generator is used to generate a set of random action values, the evaluator generates the reward values corresponding to the actions based on the designed reward function, and the random number generator generates a random number to determine the method of action selection. In this random strategy, each time the action generator is called, it generates a set of random action values that fall within the limited range. The evaluator calculates the corresponding reward values based on the current position and the random actions. The generated random action values and reward values will be stored as action–reward pairs. During the exploration process, each time an action selection is made, the system will generate an action–reward library composed of 50 groups of random action–reward pairs, as well as a random number. The choice of using 50 action–reward groups was made empirically. We found through preliminary testing that 50 samples provided a good balance between exploration diversity and computational efficiency. A larger number would increase processing time during each selection round, while a smaller number could reduce the probability of including high-reward actions in the candidate set. Therefore, 50 was selected as a practical trade-off, and this remained consistent across all experiments to ensure comparability. By comparing the random number with the preset threshold size, the system will determine which action selection method to adopt: if the random number is greater than the set threshold, the action with the highest reward will be selected; otherwise, an action will be extracted from the action reward library with a uniform probability.

The random number threshold set in this paper is 0.65. The higher the threshold, the stronger the randomness of the system's exploration of the environment. Furthermore, in this paper, the concept of experience pool is introduced during the exploration process to record each executed action and the position reached after the execution of the action. To a certain extent, it can avoid repetitive selective exploration of the same position and action, thereby improving the exploration efficiency of the system in the environment. Based on the proposed random exploration strategy, the system can explore the environment or workspace to the greatest extent, collect sufficient and rich multimodal data samples, and ensure the acquisition of a certain number of high-reward actions, thereby improving the stability and generalization ability of the multimodal network. The random exploration strategy designed in this paper is shown in Figure 2.

During the dataset collection process, the robot starts from any point in the workspace and collects the multimodal dataset at each step based on the random exploration strategy, including five kinds of modal data at the current moment—RGB images, depth images, robot pose information, contact force information, and execution action information—and four kinds of modal data for the new state reached after performing the action, namely RGB images, depth images, robot pose information, and contact force information. Among them, the resolution of the original RGB image and depth image is 640 × 480. Firstly,

480 × 480 RGB images and depth images are obtained through cutting and extraction, then 128 × 128 RGB images and depth images are obtained through scaling for storage. The pose information of the robot is directly collected from the controller, while the execution action information is obtained by calculating the pose changes of the robot before and after the movement to obtain the real motion information. The contact force information is the true contact force fed back by the environment after gravity compensation. Furthermore, in order to obtain more high-quality multimodal data samples, in addition to the random exploration scenarios within the workspace, this paper also sets up two scenarios of adjustment around the assembly hole and within the hole to increase the number of valid data samples in the assembly task.

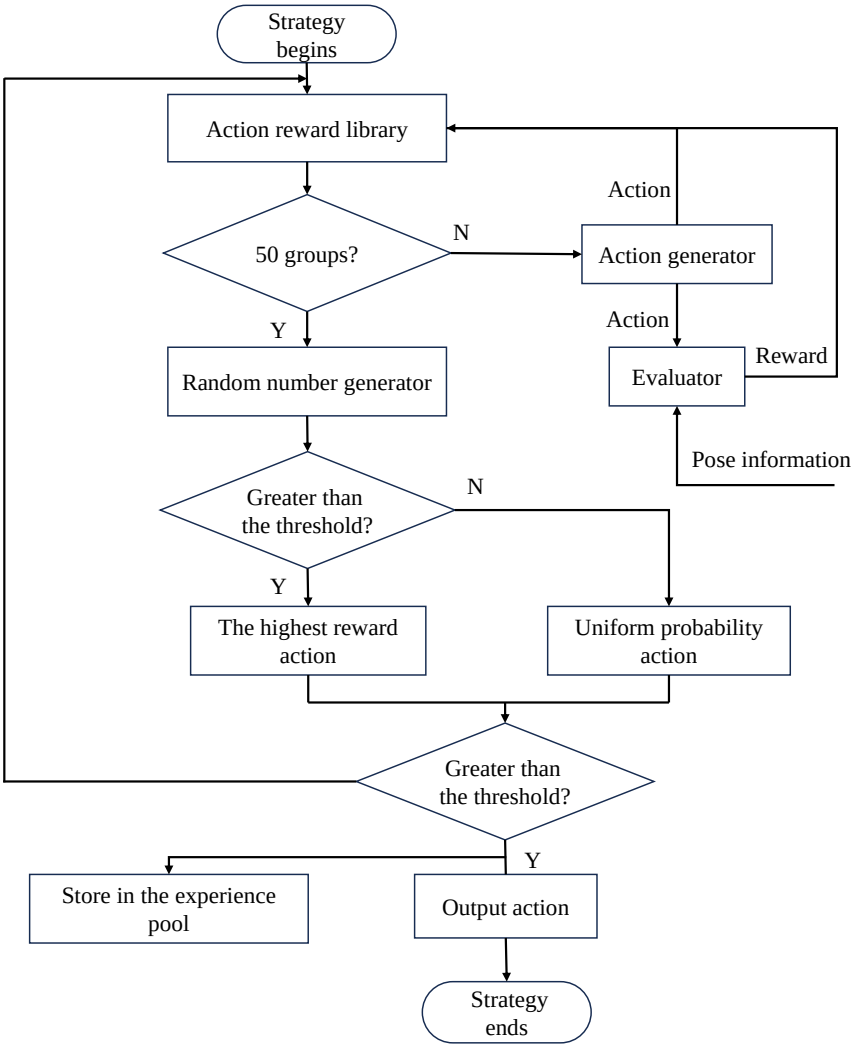


Figure 2. Single execution process of reward-based random exploration strategy. Note: The 'Pose information' in the figure is obtained from the robot's internal state feedback at each time step and is used as part of the multimodal input. It is not an external input but a measured signal during task execution.

2.3. Sac-Based Assembly Skill Learning

Once the robot can perceive its environment effectively through multimodal fusion, the next challenge is to enable the robot to learn an optimal assembly strategy. This is particularly difficult in uncertain environments where the robot needs to adapt to dynamic changes during assembly tasks. To achieve this, we use Soft Actor–Critic (SAC), a reinforcement learning (RL) algorithm that is particularly well-suited for continuous action spaces and environments with high uncertainty.

2.3.1. Soft Actor–Critic (SAC) Algorithm Overview

The Soft Actor–Critic (SAC) algorithm is a reinforcement learning algorithm in the continuous action space based on the maximum entropy theory. It enhances exploratory and robust performance by maximizing the combination of the expected return of the strategy and entropy, and it can better address the balance issue between exploration and learning. Furthermore, as an offline learning strategy, the SAC algorithm has a high sample efficiency and is capable of handling data from different sources and making more effective use of the empirical data collected during the exploration process, thereby accelerating the strategy learning process. Therefore, in this section, the SAC algorithm is adopted as the reinforcement learning strategy to learn and acquire assembly skills. The input state is the multimodal feature representation obtained by combining the contact force/torque, and the output action is the robot pose adjustment quantity. For the peg-in-hole assembly task, the pose adjustment amount output by the strategy is processed by the trajectory generator and then input into the admittance controller to generate compliant motion to the environmental contact force, ensuring the safety of the operation process. In summary, this section proposes a robot assembly skills learning strategy that integrates visual–tactile information, as shown in Figure 1.The visualization of the SAC algorithm is shown in Figure 3.

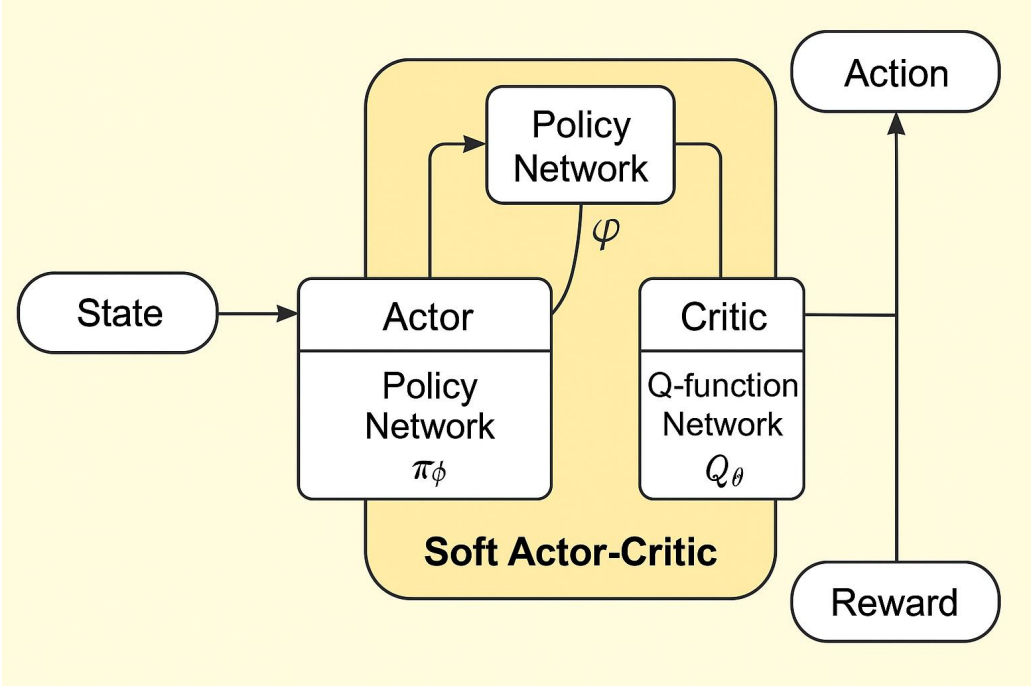


Figure 3. Visualization process of SAC algorithm.

The SAC algorithm optimizes the strategy parameters by maximizing the combination of the expected return and entropy of the strategy, which can improve the learning speed while avoiding falling into local optimal solutions. Specifically, the learning objective of the SAC algorithm is to maximize the objective function as shown in Formula (1) in order to obtain the desired optimal strategy.

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_t R(s_t,a_t) + w\mathcal{H}(\pi(\cdot \mid s_t))\right] \quad (1)$$

In the formula, π^* represents the optimal strategy; $\mathbb{E}$ represents mathematical expectation; s_t,a_t represent the state–action pairs of the agent at time t, respectively; ρ_π represents

the distribution of trajectories (s_t, a_t) under the strategy; $R(s_t, a_t)$ indicates the reward value taken a_t by the agent below s_t; and w represents the temperature coefficient, which determines the proportion of the entropy value relative to the reward. $\mathcal{H}(\pi(\cdot \mid s_t))$ represents the entropy value of the strategy π. The calculation formula for the entropy value is as follows:

$$\mathcal{H}(\pi(\cdot \mid s_t)) = -\mathbb{E}_{a\sim\pi(\cdot|s_t)}[\log \pi(a \mid s_t)] \tag{2}$$

The magnitude of the entropy value reflects the randomness degree of the strategy π. Its maximization can prevent the agent from converging too quickly to the local optimal solution, increase the exploration rate of the agent, and maintain better robustness. The SAC algorithm contains three neural network parameters, namely the policy network parameter ϕ, the value network parameter θ, and the target value network parameter θ'. The update strategy network parameters ϕ need to minimize the objective:

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim\mathcal{D},\, a_t\sim\pi_\phi}\left[\log \pi_\phi(a_t \mid s_t) - \frac{1}{w} Q_\theta(a_t \mid s_t)\right] \tag{3}$$

In the formula, $J_\pi(\phi)$ represents the minimization objective function of the network parameter ϕ; $\mathcal{D}$ represents the experience replay pool; $Q_\theta(a_t \mid s_t)$ represents the output value of the value network; and $\pi_\phi(a_t \mid s_t)$ represents the probability of the policy network's state s_t output action a_t. Action a_t can be obtained through the reparameterization method:

$$a_t = f_\phi(\epsilon_t; s_t) = \tanh\left(f_\phi^\mu(s_t) + \epsilon_t \odot f_\phi^\sigma(s_t)\right) \tag{4}$$

In the formula, $f_\phi^\mu(s_t)$ represents the average value of the output of the policy network; $f_\phi^\sigma(s_t)$ represents the variance of the output of the value network; and ϵ_t represents random noise sampled from the standard normal distribution. In the peg-in-hole task, the agent action—that is, the robot pose adjustment amount—is usually limited within a certain range. Therefore, the function tanh is used to map the action to the interval $(-1, 1)$. After reparameterization ϕ, the update of network parameters needs to minimize the objective:

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim\mathcal{D},\, a_t\sim\pi_\phi}\left[\alpha \log \pi_\phi(f_\phi(\epsilon_t; s_t) \mid s_t) - Q_\theta(f_\phi(\epsilon_t; s_t) \mid s_t)\right] \tag{5}$$

For the parameter θ of the value network, the update is carried out by minimizing the Bellman error:

$$J_Q(\theta) = \mathbb{E}_{(s_t, a_t, s_{t+1})\sim\mathcal{D},\, a_{t+1}\sim\pi_\phi}\left[\frac{1}{2}(Q_\theta(s_t, a_t) - y)^2\right] \tag{6}$$

$$y = r(s_t, a_t) + \gamma\, \mathbb{E}_{a_{t+1}\sim\pi_\phi}\left[Q_{\theta'}(s_{t+1}, a_{t+1}) - \alpha \log \pi_\phi(s_{t+1} \mid a_{t+1})\right] \tag{7}$$

In the formula, $J_\pi(\phi)$ and y represent the update functions of the network parameter θ; $s_{(t+1)}$ and $a_{(t+1)}$ represent the state–action pairs of the agent at moment $t+1$, respectively; $r(s_t, a_t)$ indicates the reward value taken a_t by the agent below s_t; and $Q_{\theta'}(s_{t+1}, a_{t+1})$ represent the output value of the target value network. For the parameter θ' of the target value network, a soft update is carried out based on Formula (4).

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{8}$$

In the formula, τ represents the update coefficient. For the temperature coefficient w, since the reward value of environmental feedback keeps changing, a changing temperature coefficient is adopted to improve the stability of the training process. The loss function of the temperature coefficient is as follows:

$$J(w) = \mathbb{E}_{a_t \sim \pi_t}[-w \log \pi_t(a_t \mid \pi_t) - w\mathcal{H}_0] \tag{9}$$

In the formula, $J(w)$ represents the loss function of the temperature coefficient, and $\pi_t(a_t \mid \pi_t)$ represent the strategy of the agent at time t.

2.3.2. Learning Process with Multimodal Representations

At each step of the assembly process, four modal data types, namely RGB images, depth images, pose information, and contact force information, are collected from the environment. Among them, RGB images, depth images, and pose information are processed using the deterministic fusion model to obtain a 128-dimensional multimodal feature representation. Then, the 128-dimensional multimodal feature representation is connected and combined with the 6-dimensional contact force/torque to form a 134-dimensional multimodal representation vector, which is input as the observed state s_t of the environment into the SAC algorithm. In the SAC algorithm, the action output of the strategy is a three-dimensional position adjustment quantity, defined as follows:

$$\mathbf{a}_t = [\Delta x, \Delta y, \Delta z] \tag{10}$$

In the formula, Δx, Δy, Δz represent the expected adjustment amounts of the assembly axis positions in the Cartesian coordinate system, respectively.

In the multimodal assembly skill learning process based on the SAC algorithm, the assembly strategy will continuously output the position adjustment amount of the assembly axis to control the continuous exploration of the environment by the assembly axis. During the exploration and training process, by maximizing the objective function shown in Formula (1) to maximize the combination of the expected return and entropy of the strategy, the problem between exploration and learning is balanced. The robot continuously learns and optimizes assembly skills in the interaction with the environment, gradually improving the assembly success rate and assembly efficiency through trial and error and feedback, solving operational problems in an uncertain environment.

2.3.3. Robot Controller Design

In this paper, a 6-dimensional vector $[x, y, z, r, p, q]$ is used to describe the pose information of the end assembly axis in the Cartesian coordinate system. Here, $[x, y, z]$ represents the position of the center at the end of the assembly axis in the robot base coordinate system, and $[r, p, q]$ represents the rotation angles around each axis of the base coordinate system; that is, the pose information. In this paper, the spatial position of the assembly peg is mainly controlled, and its posture is kept consistent with that of the assembly hole. The controller is divided into three parts: the trajectory generator, the admittance controller, and the action selector. Firstly, the SAC strategy continuously outputs the position adjustment Δx, Δy, Δz amount of the end assembly axis in the Cartesian coordinate system at a certain frequency. Secondly, the position adjustment amount is input into the trajectory generator to generate the expected motion trajectory in order to compensate for the delay difference between the output of the low-frequency strategy and the high-frequency motion control. Then, the position adjustment amount is corrected by using the admittance control based on the contact force information fed back in real time from the environment so that the robot movement has compliance during the contact process [22–28]. Finally, position control or attitude control is selected and executed through the action selector. Since the work in this paper does not involve attitude deviations, position control is always executed in the loop.

In our framework, although the high-level control policy is learned through the SAC reinforcement learning algorithm, the actual execution of motion still involves a trajectory generator. This component smooths the discrete position adjustment outputs from the

policy network to produce continuous motion profiles. Such post-processing is essential for executing learned policies on real robotic systems, where abrupt movements may lead to instability or hardware damage. Trajectory planning has traditionally been handled using minimum-jerk or piecewise polynomial interpolation methods, which emphasize kinematic smoothness and motion efficiency [29,30]. While our SAC-based method does not explicitly optimize for jerk or smoothness in the reward function, it indirectly benefits from the smoothing effect of the trajectory generator [31–36], thereby maintaining acceptable motion efficiency during the assembly process.

2.4. Integration of Visual–Tactile Fusion and SAC Learning

The integration of the visual–tactile fusion network and SAC-based skill learning forms the core of our methodology. The fusion network provides the robot with an accurate and robust perception of its environment, while the SAC algorithm enables it to learn adaptive assembly skills. Together, these components allow the robot to perform peg-in-hole assembly tasks with high success rates and strong adaptability in uncertain environments.

When applied, this integrated system enables robots to handle dynamic assembly tasks efficiently, even in the presence of environmental uncertainty and variations in object shape. The combination of vision and tactile feedback improves the robot's sensitivity to environmental changes, while SAC ensures that it can adapt to these changes and optimize its actions in real time.

3. Results and Discussion

In this section, we present a comprehensive experimental setup and results analysis to validate the effectiveness of the proposed visual–tactile fusion network and SAC-based assembly skill learning approach. Our goal is to demonstrate that the integrated system can improve robot performance in peg-in-hole assembly tasks under uncertain environments, characterized by pose deviations, environmental noise, and various peg shapes.

3.1. Experimental Setup

Gazebo software (version 11) is a commonly used simulation platform. It can combine Open Dynamics Engine (ODE) and Bullet to achieve highly realistic physical simulations, making the built simulation environment and the executed operations closer to the real scene, which is conducive to improving the reliability of the simulation results. In addition, the Gazebo platform offers a rich API and plugin system, and it supports the addition of various types of sensors (such as cameras, six-dimensional force sensors, etc.), capable of simulating complex perceptual control tasks. Therefore, in this paper, the Gazebo platform is chosen to build a set of simulation scenarios for the peg-in-hole assembly task oriented to visual and tactile perception in order to complete the simulation test of the operation task and verify the validity of the proposed network model.

(1) Task scene construction: During the process of scene construction, this paper uses the ROS-Industrial Universal Robot function package officially provided by Universal Robots as the basis, and it adopts XACRO files to describe the robot model. In the peg-in-hole assembly scenario, the peg length is 46 mm, and the hole depth is 50 mm. A total of four different shaped peg-in-hole assembly tasks, namely circular, triangular, square, and hexagonal, were designed; the gap between each group of assembly pegs and assembly holes was approximately 2 mm. The initial position of the assembly pegs was a random point in the working space. The unidirectional deviation along the X, Y, and Z axes was greater than 60 mm, and the posture remained fixed. On this basis, this paper expands the robot model, mainly including three aspects: force/torque sensor, peg-in-hole components for assembly tasks, and camera model. For the force/torque sensor and the peg-in-hole

assembly components, in this paper, the SolidWorks software (version 2021) is used to draw the model, and its STL file and physical parameters are exported for accurately describing the model in the simulation environment. In the part of the camera model,we selected the Kinect V1 camera model and fixed it in the simulation scene to comprehensively capture the entire workspace without contacting or colliding with any components in the scene.

The dataset used in this study was generated entirely within the Gazebo simulation environment under randomized initial conditions. Each episode began with a randomized peg pose (position deviation > 60 mm along X/Y/Z axes) and randomly selected peg shape (square, circle, triangle, hexagon). For each step, the robot collected a set of multimodal sensory data: RGB images, depth images, 6-DOF pose, and contact force/torque feedback. In total, we collected 110K samples, of which 80% were used for training and 20% for testing. Data preprocessing included image cropping and resizing (to 128 × 128), force normalization, and pose transformation from the robot base to end-effector frame. These steps ensure consistent input dimensionality and training stability.

(2) Sensor Configuration: The ROS-Industrial Universal Robot function package is a robot description file built for the Gazebo platform, containing basic information of the robot and rich perception options, such as joint angle state, speed state, acceleration state, joint force state, etc. Furthermore, the Gazebo platform provides a wide range of sensor API interfaces. In this paper, visual sensors and six-dimensional force/torque sensors are adopted to implement various modal environmental perception methods. The visual sensor part publishes the original RGB information and depth map information with a resolution of 640 × 480 at a frequency of 30 Hz, providing visual guidance and feedback information for robot operation. The force/torque sensor is installed between the center of the robot flange and the assembly peg, and it releases the contact force and torque between the robot and the environment at a frequency of 1000 Hz, providing force feedback information for the robot operation.

(3) Controller design: In the combination of the Gazebo platform and ROS, the ROS controller plays a key role. It assigns a ROS controller to each moving joint and moving platform component in the simulation model. These controllers are responsible for driving the virtual motors defined in the model file, thereby achieving the control of the movement of the robot model. This paper realizes the state perception and control of the robot body based on the method of perceiving and controlling the joint angle in order to provide stronger control flexibility and fully consider the contact force factor in the assembly process. During the simulation process, Gazebo releases the joint angle state of the robot in real time. In this paper, the current pose information is obtained through the solution of forward kinematics, and the joint angle information under the expected pose state is calculated based on kinematics. In terms of the underlying controller, this paper adopts the joint angle control module in the ROS-Industrial Universal Robot function package and realizes the motion control of the robot through topic release of joint angles.

(4) Evaluation Indicators: We collected the statistical data of four types of tasks in the test round, including (1) Successful assembly: when the depth of the peg is greater than 30 mm and the contact force is less than 2 N; (2) Insertion into the hole: The assembly peg is inserted into the hole but does not reach the bottom; (3) Contact assembly hole: The assembly peg is close to the assembly hole and in contact with the worktable, but fails to be inserted into the hole; and (4) Assembly failure: The assembly holes were not close to the worktable and the assembly holes. In this section, the number of adjustment steps, the depth of the peg, and the cumulative reward are used as three indicators to evaluate the quality of the peg-in-hole assembly process. In summary, in this paper, the performance of the assembly strategy is statistically analyzed based on seven types of data: the number of successful assemblies, the number of insertable holes, the number of contacts with the

worktable, the number of assembly failures, the number of adjustment steps, the insertion depth, and the cumulative reward. The reward functions adopted in this chapter are as follows:

$$r(s) = \begin{cases} v - d + h, & \text{if } h < 0 \\ v - d + 2 \times h, & \text{if } h \geq 0 \end{cases} \tag{11}$$

$$d = \left\| (P_x, P_y, P_z) - (H_x, H_y, H_z) \right\|_2 \tag{12}$$

$$h = H_z - P_z + 30 \tag{13}$$

In the formula, v is the artificially given weight parameter to achieve the reward classification; d represents the distance from the target point; and "h" represents the insertion depth. (P_x, P_y, P_z) and (H_x, H_y, H_z) represent the position of the peg-in-hole assembly and the target position in the assembly hole, respectively.

This reward formulation is designed to enhance robustness under uncertain conditions. The term d captures the Euclidean distance between the current peg tip and the target hole position, making the reward sensitive to positional perturbations caused by initialization errors or sensor noise. The insertion depth h reflects the vertical progress of insertion and penalizes shallow or misaligned attempts, which commonly occur under random initial pose offsets. Therefore, although the reward function does not explicitly encode disturbance terms, it indirectly penalizes misalignment and shallow insertion, which are primary consequences of uncertain environment factors such as initial position deviation, shape mismatch, and sensor noise. This encourages the policy to learn behavior that is robust to such variations.

By continuously evaluating d and h, the reward function guides the policy to minimize alignment errors and maximize insertion completeness, even in the presence of initial pose deviations, shape mismatch, or depth noise. These variables serve as indirect yet consistent indicators of system performance under perturbations.

Based on the above model description, sensor configuration, and controller design content, the simulation scene built on the Gazebo platform is shown in Figure 4. Furthermore, in order to verify the effectiveness for the multimodal peg-in-hole assembly task scenario, basic tests were conducted in the established simulation scenario, including tests such as the control function of the robot, the perception function of different sensors, gravity compensation, and the simulation of the contact state during the peg-in-hole process.

3.2. Experimental Verification

This section presents the verification results of the methods proposed in this paper, including the multimodal fusion module and the robot peg-in-hole task.

3.2.1. Experimental Verification of Deterministic Model

To verify the effectiveness of the visual and tactile fusion network based on the autoencoder, in this section, a robot peg-in-hole assembly simulation platform for multimodal perception will be built based on the Gazebo platform, and a multimodal dataset will be constructed by adopting the reaction-based random exploration strategy proposed in Section 2.2.3.

In the simulation platform, a total of 110K sets of multimodal data were collected based on the reward random exploration strategy, among which 80% were used for network training and 20% for network testing. Then, the network performances of the deterministic model are analyzed on the constructed multimodal dataset.

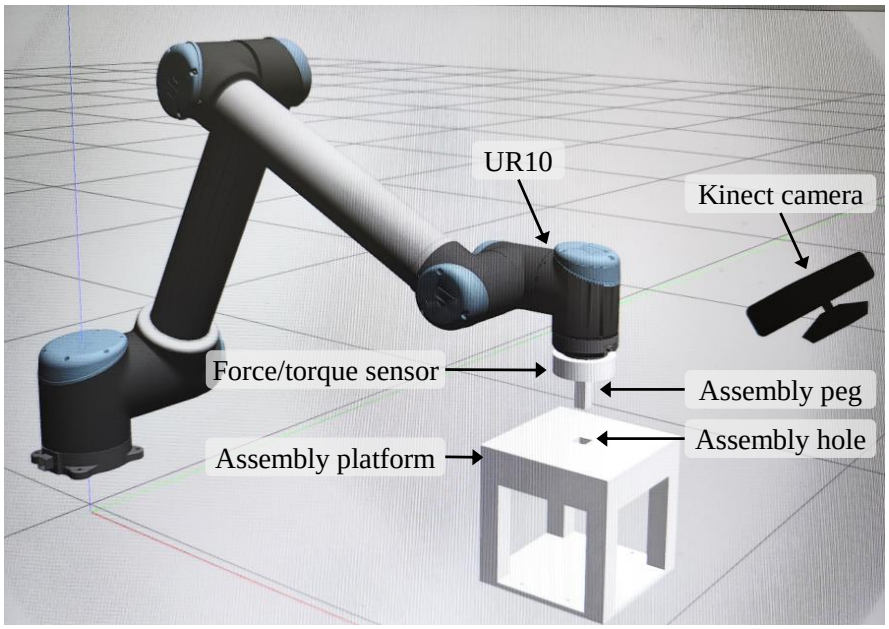


Figure 4. Robot peg-in-hole assembly simulation platform based on visual and tactile perception.

The test result of the multimodal fusion network based on the deterministic model is given as the prediction loss curve shown in Figure 5, which includes four loss situations: total prediction loss, RGB prediction loss, depth prediction loss, and pose prediction loss. Among them, the total prediction loss is the sum of the other losses. It can be observed that with the increase in the number of training sessions, all the losses of the network monotonically decrease, and convergence is achieved in the 10th episode. The loss values after network convergence are stable as shown in Table 1. The total prediction loss is 1.5×10^{-3} the RGB prediction loss is 8.7×10^{-4}, the depth prediction loss is 6.3×10^{-4}, and the pose prediction loss is 1.05×10^{-5}. The test results prove the effectiveness of the multimodal fusion network based on the deterministic model.

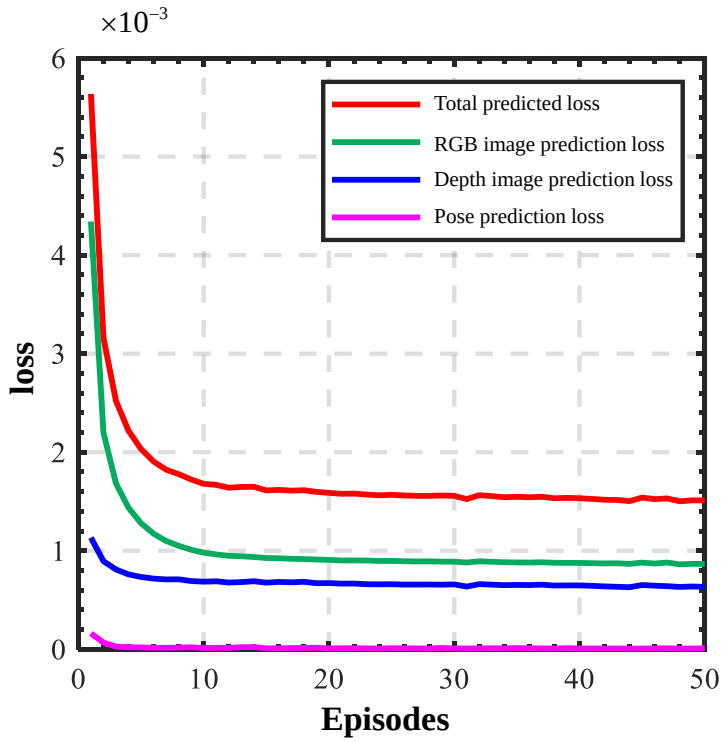


Figure 5. Prediction loss of deterministic fusion model.

Table 1. Prediction loss of deterministic fusion model.

Parameter	Loss Value
Total predicted loss	1.5×10^{-3}
RGB prediction loss	8.7×10^{-4}
Depth prediction loss	6.3×10^{-4}
Pose prediction loss	1.05×10^{-5}

3.2.2. Training and Verification of Peg-in-Hole Strategies

To visualize the learning behavior of the SAC algorithm, we present training curves that track reward accumulation, insertion accuracy, and adjustment step counts across episodes (Figures 6 and 7). These curves demonstrate the progressive improvement and convergence of the learned policy during reinforcement learning. The visualization of the soft actor–critic algorithm is shown in Figure 3.

The multimodal assembly skill learning and training based on the SAC algorithm was carried out in the simulation environment. The cumulative rewards, adjustment steps, and insertion depth results of the training process are shown in Figure 6. During training, we applied the standard SAC loss functions: a policy loss that maximizes expected Q-values with entropy regularization, a critic loss that minimizes Bellman error, and a temperature loss to tune the entropy coefficient. These losses are detailed in Equations (3)–(7) in Section 2.3.1, and their optimization enables stable learning of insertion policies. Combining the result curves of cumulative rewards and the insertion depth, it can be seen that the assembly strategy successfully completed the assembly task for the first time in the 50th episode. Subsequently, due to the random exploration actions, the test results showed a fluctuating decline, but the low frequency had little impact on the learning process. Finally, the strategy gradually converges after the 220th episode. Its cumulative reward stabilizes around a fixed value, and the assembly task is completed with a relatively high success rate, maintaining the continuous success of the operation task. By observing the adjustment step curve, it can be found that the result curve of the adjustment steps fluctuates greatly. This is caused by the large working space of the peg-in-hole assembly task and the inadequate exploration of the environment. The farther the position is from the assembly hole, the more adjustment steps are required.

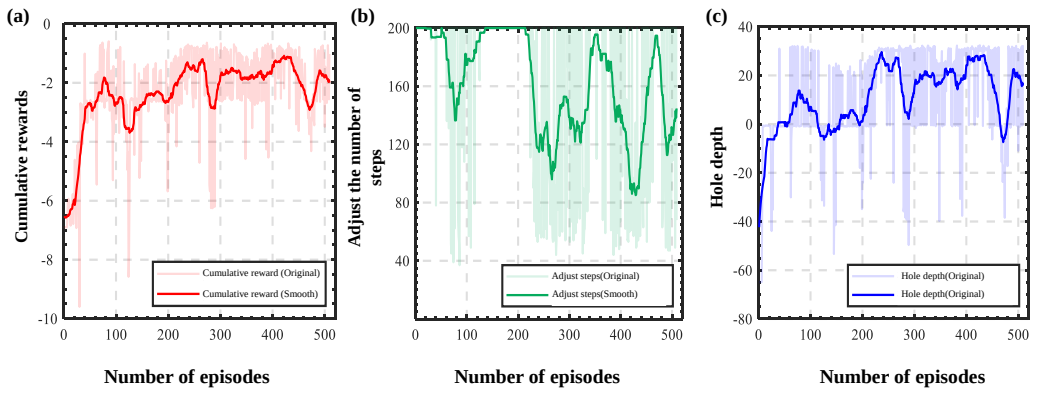


Figure 6. Learning process of peg-in-hole assembly skills based on deterministic model: (**a**) Cumulative reward per episode; (**b**) Number of adjustment steps; (**c**) Insertion depth during training.

In order to verify the effectiveness of the multimodal assembly skills based on the SAC algorithm, a random position was selected from the workspace as the initial position, and 50 sets of repetitive assembly experiments were conducted at this initial position.

The test results are shown in Figure 7. The cumulative rewards remain stable within a certain range. The fastest step adjustment to complete the assembly task is around 50 steps, and the slowest is around 150 steps. It can be seen from the the insertion depth curve that all 50 test episodes reached the expected position and successfully completed the assembly task, and the overall assembly success rate was 100%. Obviously, the assembly skill learning strategy based on the SAC algorithm can effectively complete the peg-in-hole assembly task.

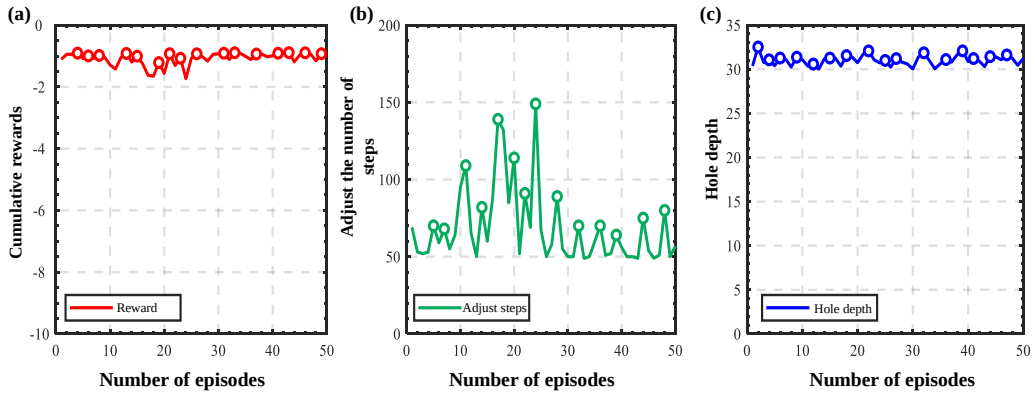


Figure 7. Assembly skill test results at the same initial position: (**a**) Cumulative reward curves; (**b**) Number of adjustment steps; (**c**) Final insertion depth in each episode.

In addition to considering the cumulative rewards, adjustment steps, and the insertion depth of the assembly process, we also recorded the contact force/torque of each step of the assembly process to analyze the contact interaction state of the robot assembly process. The experimental record of 1 episode was extracted from the above 50 test episodes. The contact force and torque situation of the assembly process based on assembly skills is shown in Figure 8.

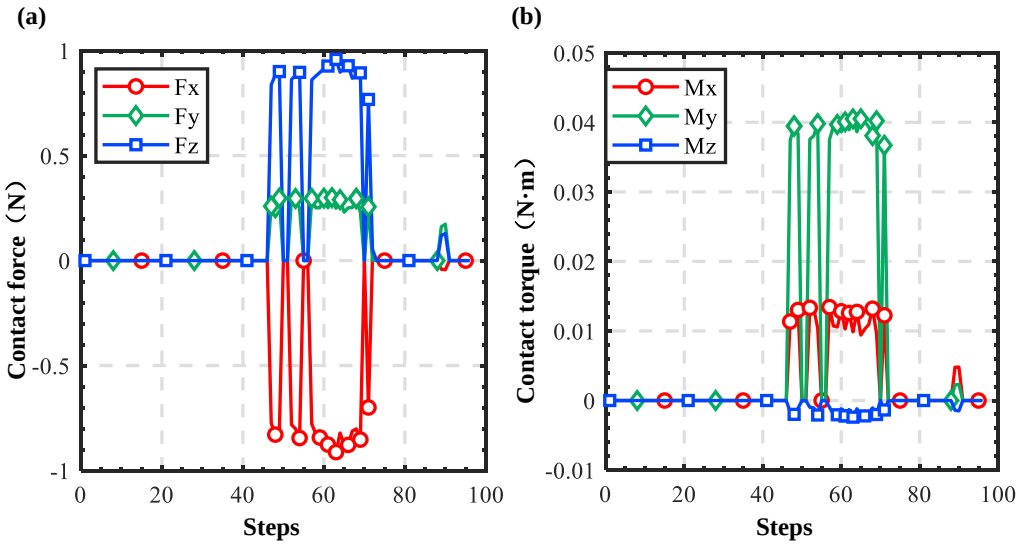


Figure 8. Contact force and torque conditions during peg-in-hole assembly: (**a**) Contact force over steps; (**b**) Contact torque over steps.

It can be seen from the process curves of force and torque that in the first 45 steps of this episode, it is in the hole search process, and the assembly peg does not come into contact with the environment, so the values of force and torque are close to 0. At step 45, it reaches the edge of the assembly hole and starts to attempt alignment and contact,

with a contact force of about 1 N. After alignment, the contact force becomes 0 and moves downward to insert it into the hole. At step 57, contact occurs with the inner wall of the assembly hole, with a contact force close to 1 N. After adjustment, it continues inserting the assembly peg into the hole to complete the assembly task. The contact force and torque during the assembly process are always at a relatively small value. The results show that the assembly skill learning strategy based on the SAC algorithm can learn a relatively safe operation trajectory.

To evaluate the efficiency of the proposed assembly strategy, we estimate the average execution time per task episode based on the control cycle and number of adjustment steps. In our simulation setup, each control action corresponds to a single adjustment step and is executed at a frequency of 10 Hz, implying that each step takes approximately 0.1 s. As shown in Figure 6, the number of adjustment steps required for successful assembly ranges from 80 to 150 steps depending on the initial distance to the target. Consequently, the estimated duration for completing a single peg-in-hole task falls between 8 and 15 s. This estimation is consistent with other simulation-based robotic assembly works using similar controllers and platforms [7,10]. Although real-world latency and mechanical constraints may further affect timing, our results demonstrate that the proposed method offers a reasonable balance between precision and speed.

3.2.3. Generalization Experiments at Different Initial Positions

In order to verify the generalization ability of the multimodal assembly skills based on the SAC algorithm, 50 episodes of assembly experiments were carried out at different initial positions to verify the performance of the proposed algorithm. Before the start of each episode, an initial position was randomly selected from the workspace (to increase the test difficulty, the selected initial positions were all far away from the assembly holes), and the assembly task began from the randomly selected initial position to test the generalization effect of the assembly skills. The results of the assembly skills generalization experiment at different initial positions are shown in Figure 9. It can be seen that the cumulative reward stabilized within a range, but its degree of fluctuation was greater than that of the test results under the fixed position. This is due to the different distances from the random initial position to the target point and the different number of adjustment steps required for each episode, which led to fluctuation in reward. In terms of the number of adjustment steps, the fastest one completed the assembly task in close to 40 steps, while the slowest one took approximately 150 steps to complete the task. Furthermore, it can be seen from the insertion depth curve that the 50 test episodes reached the target position and successfully completed the assembly task, and the overall assembly success rate was 100%. The test results show that the assembly skills learned based on the SAC algorithm can be effectively generalized to different initial positions.

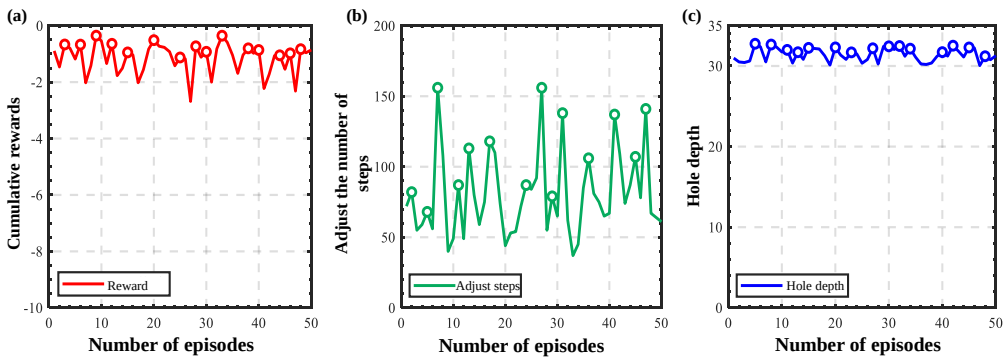


Figure 9. Assembly skill generalization experiment under different initial positions: (**a**) Cumulative rewards; (**b**) Number of adjustment steps; (**c**) Final insertion depth.

3.2.4. Generalization Experiments for Different Types of Holes

In order to verify the generalization ability of the multimodal assembly skills based on the SAC algorithm, the assembly skills learned from the square peg-in-hole assembly scenario were applied to the assembly scenarios of different types of pegs-in-holes for generalizability testing to verify the performance of the proposed algorithm. Meanwhile, consistent with the generalization experiments at different initial positions, in each episode of this experiment, a target point was randomly selected from the workspace as the initial position. A total of 50 episodes of assembly tests were conducted, and the cumulative rewards, adjustment steps, and insert depth results during the test process were recorded.

The generalization experiment results of the square peg-in-hole strategy applied to the circular peg-in-hole assembly scenario are shown in Figure 10. It can be seen that the cumulative reward was stable within a range, and the number of adjustment steps fluctuated greatly due to the influence of the initial position deviation. However, the success rate of 50 test episodes remained at 100%.

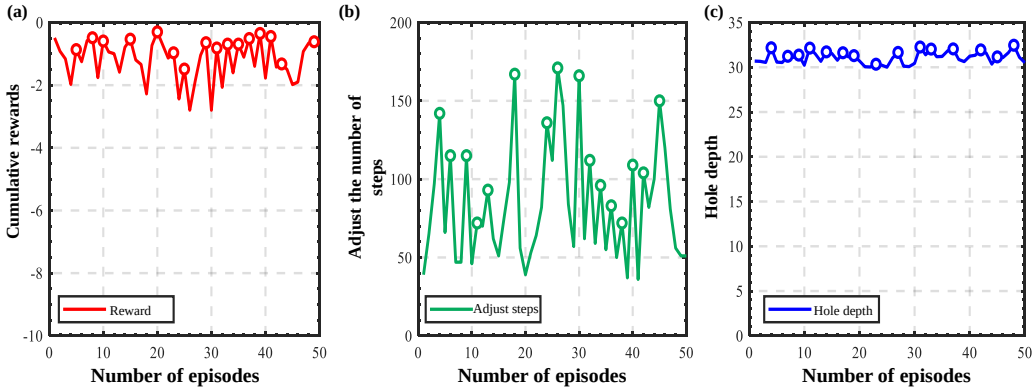


Figure 10. Generalization experiment of square peg-in-hole strategy applied to circular hole assembly: (**a**) Cumulative rewards; (**b**) Number of adjustment steps; (**c**) Final insertion depth.

The results of the generalization experiment of the square peg-in-hole strategy applied to the triangular peg-in-hole assembly scenario are shown in Figure 11. It can be seen that the cumulative reward can still remain stable within a range. However, in this experiment, the assembly was not completed, with a success rate of 84%. Among them, the assembly tasks were successfully completed in 42 episodes. Two episodes were inserted into the holes (18 mm and 29 mm were inserted, respectively), and six episodes were in contact with the assembly holes. No assembly failure occurred. The episodes of incomplete assembly tasks and assembly failures were mainly due to the interruption of the assembly task caused by the program after the adjustment steps reached 200.

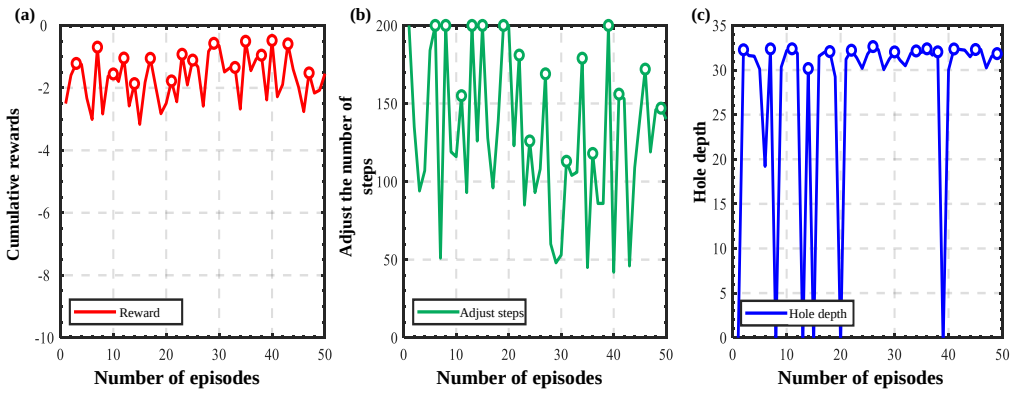


Figure 11. Generalization experiment of square peg-in-hole strategy applied to triangular hole assembly: (**a**) Cumulative rewards; (**b**) Number of adjustment steps; (**c**) Final insertion depth.

The generalization experimental results of the square peg-in-hole strategy applied to the hexagonal peg-in-hole assembly scenario are shown in Figure 12. It can be seen that the cumulative reward can still remain stable within a range. Moreover, in this experiment, all the assembly tasks in all episodes were inserted into the holes. Among them, the assembly tasks were successfully completed in 49 episodes, and 14 mm was inserted into the hole in 1 episode (the adjustment steps exceeded the limit). The success rate was 98%.

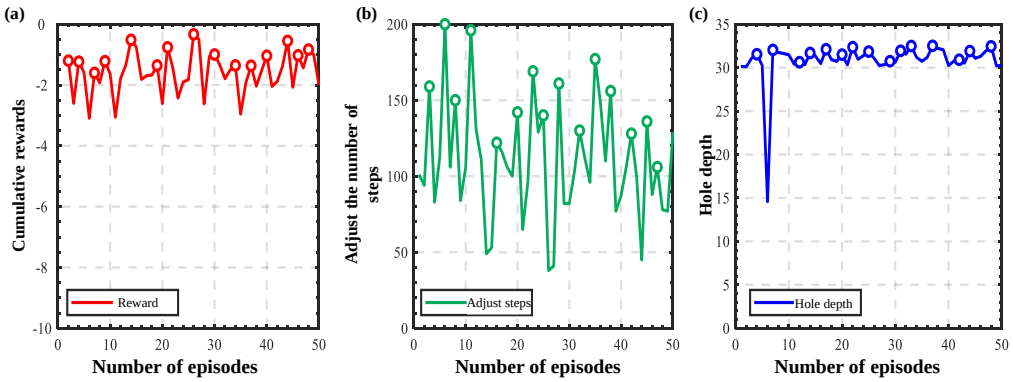


Figure 12. Generalization experiment of square peg-in-hole strategy applied to hexagonal hole assembly: (**a**) Cumulative rewards; (**b**) Number of adjustment steps; (**c**) Final insertion depth.

By analyzing the above test results, it can be seen that the multimodal assembly skills based on the SAC algorithm can be effectively generalized to the assembly scenarios of different types of peg-in-hole tasks. As for the episodes that failed to successfully complete the assembly tasks, the fundamental reason might be insufficient training in operation strategies and an incomplete understanding of the environment.

Compared to traditional model-based insertion methods or visual-only policies, our framework achieves comparable or superior performance in terms of insertion accuracy. For instance, prior visual-guided methods under similar conditions often report success rates in the range of 80–95% [7]. In contrast, our SAC-based multimodal policy maintained 100% success in square and circular scenarios and over 98% in generalization tasks including triangular and hexagonal peg types, as shown in Figure 12. This highlights the improved robustness and generalization enabled by the proposed fusion strategy [37–40].

3.2.5. Comparison with Existing Approaches

Compared to conventional vision-only insertion strategies [7,8], our method significantly improves robustness by incorporating real-time tactile feedback and proprioception. While vision-based approaches offer good spatial awareness, they often fail under occlusions, reflective surfaces, or minor pose deviations. Tactile-guided methods [9,41] improve local contact control but lack global perception. In contrast, our multimodal fusion network encodes RGB, depth, and pose signals into a unified latent representation, enabling the policy to reason jointly over visual and tactile cues. Furthermore, by integrating the Soft Actor–Critic (SAC) algorithm, the system learns adaptive control policies that generalize across different shapes and initial conditions.

Recent works using reinforcement learning for peg-in-hole tasks [10,42] typically focus on unimodal data or require extensive domain randomization. Our method stands out by leveraging structured feature fusion, allowing for efficient policy training and strong generalization without additional demonstrations or handcrafted controllers.

4. Conclusions

This paper introduces an integrated approach combining visual–tactile fusion with the Soft Actor–Critic (SAC) algorithm to improve the robustness and adaptability of robots performing peg-in-hole assembly tasks in uncertain environments. The proposed multimodal fusion network enhances the robot's ability to perceive its environment by combining visual, depth, tactile, and pose information, thus improving its accuracy in detecting object alignments and adjusting forces during the assembly process. The SAC-based learning model further empowers the robot to learn assembly strategies that generalize well across different conditions, enabling the system to handle variations in peg shapes and environmental factors.

Our experimental results highlight the effectiveness of the combined approach. The fusion network improves the robot's performance, reducing misalignments and enhancing force control, while the SAC algorithm enables fast adaptation to new tasks and better generalization to different peg shapes and poses. Additionally, the results indicate that the integrated system performs robustly in the presence of environmental noise and dynamic changes, providing a stable solution for industrial assembly operations. These experimental results further validate the system's robustness across uncertain conditions, including variations in initial pose, contact dynamics, shape diversity, and sensor noise. Such resilience is essential for real-world deployment in dynamic industrial environments.

In summary, this work contributes to the field of robotic assembly by offering a scalable, adaptive solution that leverages advanced perception and learning techniques. Future work could explore further enhancements to the fusion network, improve learning efficiency with more complex environments, and expand the application of this method to other types of assembly tasks and industrial applications. The approach holds promise for advancing automation in manufacturing and other industries.

Author Contributions: Conceptualization, J.T. and X.Y.; Methodology, J.T. and X.Y.; Validation, J.T., X.Y. and S.L.; Writing—original draft, J.T.; Writing—review & editing, S.L.; Supervision, S.L.; Funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of Guangxi-General Program under Grant 2025GXNSFAA069931, and in part by the Natural Science Foundation of Guangxi-Young Scientists Fund under Grant 2023GXNSFBA026069.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Shen, L.; Su, J.; Zhang, X. Review on Peg-in-Hole Insertion Technology Based on Reinforcement Learning. In Proceedings of the 2023 China Automation Congress (CAC), Chongqing, China, 17–19 November 2023; pp. 6688–6695.
2. 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.
3. Zhang, X.; Sun, L.; Kuang, Z.; Tomizuka, M. Learning Variable Impedance Control via Inverse Reinforcement Learning for Force-Related Tasks. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2225–2232. [CrossRef]
4. Jiang, J.; Huang, Z.; Bi, Z.; Ma, X.; Yu, G. State-of-the-art control strategies for robotic PiH assembly. *Robot. Comput.-Integr. Manuf.* **2020**, *65*, 101894. [CrossRef]
5. Sun, T.; Liu, H. Adaptive force and velocity control based on intrinsic contact sensing during surface exploration of dynamic objects. *Auton. Robot.* **2020**, *44*, 773–790. [CrossRef]
6. Yan, S.; Xu, D.; Tao, X. Hierarchical policy learning with demonstration learning for robotic multiple peg-in-hole assembly tasks. *IEEE Trans. Ind. Inform.* **2023**, *10*, 10254–10264. [CrossRef]

7. Kim, B.; Choi, M.; Son, S.; Yun, D.; Yoon, S. Vision-force guided precise robotic assembly for 2.5D components in a semistructured environment. *Assem. Autom.* **2021**, *41*, 200–207. [CrossRef]
8. Xu, J.; Liu, K.; Pei, Y.; Yang, C.; Cheng, Y.; Liu, Z. 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*, 3509815. [CrossRef]
9. Zou, P.; Zhu, Q.; Wu, J.; Xiong, R. Learning-based Optimization Algorithms Combining Force Control Strategies for Peg-in-Hole Assembly. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 7403–7410.
10. Schoettler, G.; Nair, A.; Luo, J.; Bahl, S.; Ojea, J.A.; Solowjow, E.; Levine, S. Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 5548–5555.
11. Dimeas, F.; Aspragathos, N. Reinforcement learning of variable admittance control for human-robot co-manipulation. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–3 October 2015; pp. 1011–1016.
12. Feng, Y.; Shi, C.; Du, J.; Yu, Y.; Sun, F.; Song, Y. Variable admittance interaction control of UAVs via deep reinforcement learning. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 1291–1297.
13. Kumhar, H.S.; Kukshal, V. A review on reinforcement deep learning in robotics. In Proceedings of the 2022 Interdisciplinary Research in Technology and Management (IRTM), Kolkata, India, 24–26 February 2022; pp. 1–8.
14. Singh, B.; Kumar, R.; Singh, V.P. Reinforcement learning in robotic applications: A comprehensive survey. *Artif. Intell. Rev.* **2022**, *55*, 945–990. [CrossRef]
15. Elguea-Aguinaco, Í.; Serrano-Muñoz, A.; Chrysostomou, D.; Inziarte-Hidalgo, I.; Bøgh, S.; Arana-Arexolaleiba, N. A review on reinforcement learning for contact-rich robotic manipulation tasks. *Robot. Comput.-Integr. Manuf.* **2023**, *81*, 102517. [CrossRef]
16. Spector, O.; Di Castro, D. InsertionNet: A Scalable Solution for Insertion. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5509–5516. [CrossRef]
17. Ma, Y.; Xie, Y.; Zhu, W.; Liu, S. An Efficient Robot Precision Assembly Skill Learning Framework Based on Several Demonstrations. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 124–136. [CrossRef]
18. Hou, Z.; Fei, J.; Deng, Y.; Xu, J. Data-efficient hierarchical reinforcement learning for robotic assembly control applications. *IEEE Trans. Ind. Electron.* **2020**, *68*, 11565–11575. [CrossRef]
19. Lee, M.A.; Zhu, Y.; Zachares, P.; Tan, M.; Srinivasan, K.; Savarese, S.; Fei-Fei, L.; Garg, A.; Bohg, J. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Trans. Robot.* **2020**, *36*, 582–596. [CrossRef]
20. Jin, L.; Men, Y.; Song, R.; Li, F.; Li, Y.; Tian, X. Robot Skill Generalization: Feature-Selected Adaptation Transfer for Peg-in-Hole Assembly. *IEEE Trans. Ind. Electron.* **2024**, *71*, 2748–2757. [CrossRef]
21. Hua, J.; Zeng, L.; Li, G.; Ju, Z. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors* **2021**, *21*, 1278. [CrossRef]
22. Whitney, D.E.; Rourke, J.M. Mechanical behavior and design equations for elastomer shear pad remote center compliances. *J. Dyn. Syst. Meas. Control* **1986**, *108*, 223–232. [CrossRef]
23. Asada, H.; Kakumoto, Y. The dynamic RCC hand for high-speed assembly. In Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA, USA, 24–29 April 1988; pp. 120–125.
24. Sturges, R.H.; Laowattana, S. Fine motion planning through constraint network analysis. In Proceedings of the IEEE International Symposium on Assembly and Task Planning, Pittsburgh, PA, USA, 10–11 August 1995; pp. 160–170.
25. Zhang, Q.; Hu, Z.; Wan, W.; Harada, K. Compliant Peg-in-Hole Assembly Using a Very Soft Wrist. *IEEE Robot. Autom. Lett.* **2023**, *9*, 17–24. [CrossRef]
26. Choi, S.; Kim, D.; Kim, Y.; Kang, Y.; Yoon, J.; Yun, D. A Novel Compliance Compensator Capable of Measuring Six-Axis Force/Torque and Displacement for a Robotic Assembly. *IEEE/ASME Trans. Mechatron.* **2023**, *29*, 29–40. [CrossRef]
27. Xu, X.; Zhu, D.; Zhang, H.; Yan, S.; Ding, H. Application of novel force control strategies to enhance robotic abrasive belt grinding quality of aero-engine blades. *Chin. J. Aeronaut.* **2019**, *32*, 2368–2382. [CrossRef]
28. Solanes, J.E.; Gracia, L.; Munoz-Benavent, P.; Esparza, A.; Miro, J.V.; Tornero, J. Adaptive robust control and admittance control for contact-driven robotic surface conditioning. *Robot. Comput.-Integr. Manuf.* **2018**, *54*, 115–132. [CrossRef]
29. Lu, S.; Zhang, X.; Wu, C.; Wang, H. Kinematics and dynamics analysis of the 3PUS-PRU parallel mechanism module designed for a novel 6-DOF gantry hybrid machine tool. *J. Mech. Sci. Technol.* **2020**, *34*, 345–357. [CrossRef]
30. Lu, S.; Zhang, X.; Wu, C.; Wang, H. Minimum-jerk trajectory planning pertaining to a translational 3-DOF parallel manipulator through piecewise quintic polynomials interpolation. *Adv. Mech. Eng.* **2020**, *12*, 168781402091366. [CrossRef]
31. Chhatpar, S.R.; Branicky, M.S. Search strategies for peg-in-hole assemblies with position uncertainty. In Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Maui, HI, USA, 29 October–3 November 2001; pp. 1465–1470.

32. Sharma, K.; Shirwalkar, V.; Pal, P.K. Intelligent and Environment-Independent Peg-In-Hole Search Strategies. In Proceedings of the 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE-2013), Jabalpur, India, 16–18 December 2013.
33. Newman, W.S.; Zhao, Y.H.; Pao, Y.H. Interpretation of force and moment signals for compliant peg-in-hole assembly. In Proceedings of the 2001 IEEE International Conference on Robotics and Automation, Seoul, Republis of Korea, 21–26 May 2001; pp. 571–576.
34. Sharma, K.; Shirwalkar, V.; Pal, P.K. Peg-In-Hole search using convex optimization techniques. *Ind. Robot. Int. J.* **2017**, *44*, 618–628. [CrossRef]
35. Chernyakhovskaya, L.B.; Simakov, D.A. Peg-on-hole: Mathematical investigation of motion of a peg and of forces of its interaction with a vertically fixed hole during their alignment with a three-point contact. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 689–704. [CrossRef]
36. Wu, W.; Liu, K.; Wang, T. Robot assembly theory and simulation of circular-rectangular compound peg-in-hole. *Robotica* **2022**, *40*, 1–34. [CrossRef]
37. Luo, Z.; Li, J.; Bai, J.; Wang, Y.; Liu, L. Adaptive hybrid impedance control algorithm based on subsystem dynamics model for robot polishing. In Proceedings of the 2019 Intelligent Robotics and Applications (ICIRA), Shenyang, China, 8–11 August 2019; Springer: Cham, Switzerlan, 2019; pp. 163–176.
38. Jin, Z.; Qin, D.; Liu, A.; Zhang, W.; Yu, L. Model predictive variable impedance control of manipulators for adaptive precision-compliance tradeoff. *IEEE/ASME Trans. Mechatron.* **2022**, *28*, 1174–1186. [CrossRef]
39. Gai, Y.; Guo, J.; Wu, D.; Chen, K. Feature-based compliance control for precise peg-in-hole assembly. *IEEE Trans. Ind. Electron.* **2021**, *69*, 9309–9319. [CrossRef]
40. Huang, J.; Chen, S.; Zheng, W.; Su, P.; Li, J.; Zheng, J.; Liang, Y.; Xiao, H.; Peng, Y.; Huang, Z. Fuzzy Adaptive Compliance Control Method for Charging Manipulator. In Proceedings of the 2023 IEEE International Conference on Mechatronics and Automation (ICMA), Harbin, China, 6–9 August 2023; pp. 992–997.
41. Lee, D.; Choi, M.; Park, H.; Jang, G.; Park, J.; Bae, J. Peg-in-Hole Assembly with Dual-Arm Robot and Dexterous Robot Hands. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8566–8573. [CrossRef]
42. Hao, P.; Lu, T.; Cui, S.; Wei, J.; Cai, Y.; Wang, S. Meta-Residual Policy Learning: Zero-Trial Robot Skill Adaptation via Knowledge Fusion. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3656–3663. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.