CIRP Annals - Manufacturing Technology 74 (2025) 13–17

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: https://www.editorialmanager.com/CIRP/default.aspx

Vision intelligence-conditioned reinforcement learning for precision assembly

Sichao Liu, Lihui Wang (1)*

Department of Production Engineering, KTH Royal Institute of Technology, Stockholm, Sweden

ARTICLE INFO

Article history:
Available online 11 May 2025

Keywords:
Robot
Assembly
Reinforcement learning

ABSTRACT

Robots that embrace human-level performance on precise, dexterous and dynamic assembly tasks can significantly enhance the efficiency in precision assembly but remain big challenges. This paper introduces a vision intelligence-conditioned method for precision assembly, enabled by human-in-the-loop reinforcement learning. Upon visual demonstrations collected and trained by a reward classifier, a data-efficient reinforcement learning algorithm trains and learns vision-based robotic manipulation policies under human-in-the-loop corrections. An impedance-based control strategy derived from policies and visual guidance achieves high-precision contact-rich assembly manipulations with near-perfect success rates (above 98%) and compliance behaviours. The effectiveness of the presented method is experimentally demonstrated with semiconductor assembly.

1. Introduction

Robot-assisted assembly with human-level performance is a long-standing challenge, especially for dynamic and precise assembly tasks [1,2]. These tasks often require sub-millilitre precision in the robot's movement and control, and they also involve so-called contact-rich manipulations (e.g., insertion and screwing), during which there are contacts between the components to be assembled [3]. The success of the task depends heavily on consistent and precise physical interactions between the robot and objects/environment. However, conventional robotic assembly methods neither support dexterous and dynamic tasks due to robots controlled pre-generated rigid codes nor learn precise manipulation/assembly policies [4,5], followed by the impacts of inefficiencies (e.g., in productivity and cost). For this purpose, reinforcement learning (RL) holds the promise of learning complex and dexterous robotic manipulation skills through trial and error [6,7], where the skills can align with the physical demands (i.e., precision) of the tasks. For example, the use of RL in robotic assembly tasks has demonstrated the capability of learning complex and various manipulation policies directly in the real world [8]. Recent advancements in vision intelligence offer the potentials to incorporate visual perception into RL for precision assembly tasks. As an example, a vision-based RL system integrates task-level visual observations to achieve dexterous robotic manipulation tasks with a high success rate [2].

RL methods have been effective for training on existing large-scale datasets for robust policies. However, this is impractical for physical robots, since sample collection costs and time constraints are significant [9]. Therefore, real-world robotic RL demands sample-efficient algorithms that can handle high-dimensional inputs (i.e., robot state and perception) and enable straightforward reward and reset specification [10]. Despite recent advances of RL algorithms, policy training often takes a long time. Also, RL struggles to work effectively with robots on its own for contact-rich manipulation tasks if an impedance controller or similar compliance mechanism is not in place, since it cannot handle the task's nuance [3]. Impedance control bridges the gap by managing low-level contact dynamics, safety, and compliance, allowing RL to focus on high-level decision-making [4]. Therefore, incorporating an impedance controller into RL enables the robots to tackle real-world manipulation tasks effectively and robustly.

This paper presents a vision intelligence-based precision assembly method supported by RL and impedance control. Upon offline demonstration collection, a pre-trained vision model-based reward function is presented to train a binary classifier. Then, under human demonstration and corrections, a sample-efficient RL based on an algorithm of RL with prior data (RLPD) [10] takes actions and state observations to train a robotic manipulation policy. Finally, an impedance-based control strategy with visual guidance and RL policies regulates robot compliance behaviours and performs precise assembly with a near-perfect success rate.

2. RL-based policy learning for precision assembly

2.1. Overview of RL policy learning for precision assembly

The goal of vision-based RL policy learning for precision assembly is to maximise the probability of success of each trajectory in completing assembly tasks using state observations and actions as inputs. As shown in Fig. 1, it starts from an offline demonstration collection of contact-rich assembly tasks by using an input device (SpaceMouse), and they are labelled as success and failure samples. These samples work as inputs to a reward function to train a binary classifier, which will guide and accelerate the policy training process. Then, a sample-efficient RL algorithm uses an actor-learner architecture for precise manipulation policy learning directly in the real world, rather than adopting a sim2real pipeline. During the process, the actor receives the updated policy parameters from the learner, interacts with the environment for task execution, and sends policy transitions (i.e., observations, actions, rewards) into the replay buffer, together with robot trajectories of human demos and interventions (accelerating training or policy correction).

* Corresponding author.
E-mail address: lihui.wang@iip.kth.se (L. Wang).

https://doi.org/10.1016/j.cirp.2025.04.016

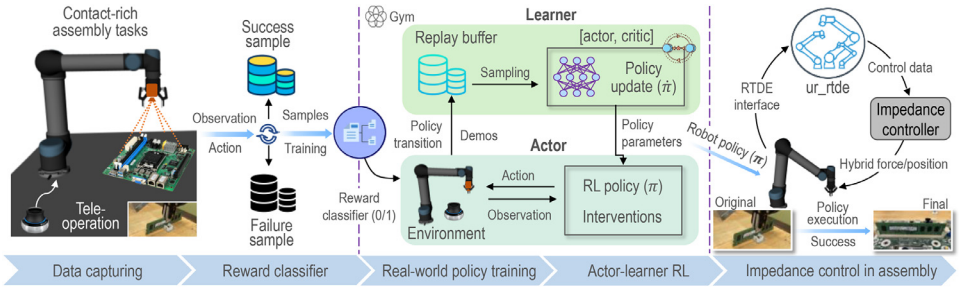

Fig. 1. Workflow of vision intelligence and reinforcement learning for precision assembly.

The learner samples from the replay buffer to update the policy parameters by using the RLPD method, where the actor-critic-based network architecture is adopted. It decides which action is taken by the actor and evaluates the action followed by action adjustment. Once the maximum steps of the training are completed, the robotic policy for contact-rich and precision assembly tasks is generated. To manage the success of the tasks, the impedance-based control strategy with visual observations is adopted to guide precise robot movement with the appropriate force that is calculated based on the difference between current and target poses. Here, an RTDE (real-time data exchange) control interface is used for data flow and command execution on a robot controller.

2.2. Vision system for demonstration collection

The system adopts and initialises an environment of Gymnasium (Gym), which is an open-source Python library for developing and comparing RL algorithms [11]. It is controlled by reset and step methods for environment reset and one-time-step running of the environment dynamics, and it takes actions as inputs and returns a tuple: (observation, reward, terminated, truncated, info). As shown in Fig. 2(a), the system with wrist-side cameras and a SpaceMouse is for visual observation and robot teleoperation, respectively. Here, the wrist cameras (RealSense D405 in precision applications) record high-resolution RGB images that are cropped as 128×128 pixels with the focus of the area of interest for computational efficiency, at a sampling rate of 30 Hz. An operator teleoperates the robot to perform the assembly tasks and collects samples, where the SpaceMouse gives a 6D Cartesian pose and gripper command to the robot. Specifically, for a task of inserting a RAM card into a slot, the sample is labelled as a *success* if the absolute value of the difference between the current and target poses is less than a threshold (0.1 mm), and as a *failure* otherwise. An RTDE control interface with a frequency of 10 Hz is adopted for real-time control and interaction.

Here, 100 success and 1300 failure data points are sampled, respectively. For each sample, robot proprioceptive state and image observations from two wrist cameras are used as input to the binary reward classifier. The RGB image is processed with a pre-trained vision model (ResNet-10) to extract spatial feature embedding. The encoders' outputs (image embedding) and any additional proprioceptive data are fed into fully-connected layers (FCLs: 3) to predict the class logits (probability of each observation in two classes). The classifier is trained with 100 epochs and an Adam optimiser. Fig. 2(b) shows the training accuracy and loss, and the results show high accuracy (almost 100 %) and efficiency (30 epochs for convergence). Finally, the checkpoints of the reward classifier include all learned weights for visual and proprioceptive encoders, which are to guide the RL policy training.

2.3. Reinforcement learning for robotic assembly tasks

A robotic RL task is defined via a *Markov Decision Process* $\boldsymbol{M} = \{\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, p, \boldsymbol{R}, \gamma\}$, where $\boldsymbol{s} \in \boldsymbol{S}$ is the state observation space (proprioceptive state and environmental observation), $\boldsymbol{a} \in \boldsymbol{A}$ is the robot action space (robot's Cartesian pose and gripper state), $\boldsymbol{P}(\boldsymbol{s})$ is a distribution over initial states, p is the transition probabilities that capture the system dynamics, R is the reward function that uses a pre-trained binary classifier to assess whether the task is successful or not, and $0 \leq \gamma < 1$ is a discount factor that determines how much weight is given to future rewards when evaluating a state or action. The goal of RL is to discover an optimal policy π that maps states to actions so as to maximise the cumulative expected value of the reward ($E[\sum_{t=0}^{h} \gamma^t R(\boldsymbol{s}_t, \boldsymbol{a}_t)]$). Specifically, it is to maximise the probability of success for each trajectory in completing precision assembly tasks in this study.

The success of RL in robotic manipulation tasks has made them effective in training a policy for precision assembly. Three criteria are considered for choosing an RL algorithm in the present study: (1) sample efficiency; (2) convergence speed, and (3) training time. To address the algorithm selection criteria, how to choose an RL algorithm is discussed in [12]. Given the assembly task with high-precision and contact-rich features, this study selects RLPD as the RL algorithm due to its sample efficiency and ability to incorporate prior data. During each training step, RLPD samples a batch (prior and on-policy data) to update the parameters of a parametric Q-function $Q_\varphi(\boldsymbol{s}, \boldsymbol{a})$ and the policy $\pi_\theta(\boldsymbol{a}|\boldsymbol{s})$ according to the gradient of their loss functions (Q-loss: $\mathcal{L}_Q$ and policy loss: $\mathcal{L}_\pi$):

$$\mathcal{L}_Q(\varphi) = E_{\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}'}\left[\left(Q_\varphi(\boldsymbol{s},\boldsymbol{a}) - (R(\boldsymbol{s},\boldsymbol{a}) + \gamma E_{\boldsymbol{a}' \sim \pi_\theta}[Q_{\bar{\varphi}}(\boldsymbol{s}',\boldsymbol{a}')])\right)^2\right] \quad (1)$$

$$\mathcal{L}_\pi(\theta) = -E_{\boldsymbol{s}}[E_{\boldsymbol{a} \sim \pi_\theta(\theta)}[Q_\varphi(\boldsymbol{s},\boldsymbol{a})] + \tau\Phi(\pi_\theta(.|\boldsymbol{s}))] \quad (2)$$

where $Q_{\bar{\varphi}}$ is a target network, and the policy loss uses entropy regularisation (Φ), controlled by the temperature parameter τ.

2.4. Actor-learner-replay-based reinforcement learning policy

An actor-learner architecture for the policy training is adopted as shown in Fig. 3, given its scalability, efficiency, and stability—especially in data-hungry or parallelisable settings [13]. It includes three core components: actor, leaner and replay buffer. The system workflow starts with the actor process, and it adopts an agent-environment loop. The actor interacts with the environment by executing the action on the robot, where the action can be intervened by tele-operating the robot if necessary or the updated policy parameters from the learner. It then sends policy transitions and/or intervention actions to the replay buffers (RL and demo buffers: storing on-policy data and offline human demos). During the training, the learner updates the policy through training batch samples evenly from two buffers. The policy may explore incorrect behaviours (i.e., moving away from the target), lead the robot to an undesirable state, and take a long time from a local optimum [2]. To tackle these challenges, dataset augmentation (Dagger) and its variants

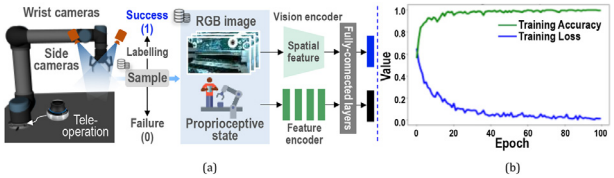

Fig. 2. (a) Workflow of a reward function classifier; (b) Training results.

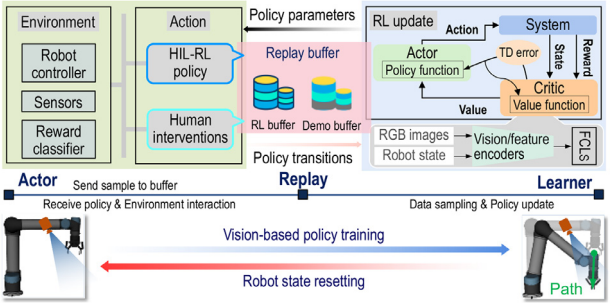

Fig. 3. Actor-learner-based RL for policy training with human interventions.

are used to incorporate human interventions in refining the policy via supervised learning [14]. Specifically, a human operator supervises the robot during the process and manipulates the SpaceMouse to correct robot behaviours when necessary. Human interventions are stored in these two buffers, which are not only for freeing the robot from stuck situations but also for speeding up the RL policy training process.

Therefore, an offline-policy RL with human-in-the-loop (HIL) intervention and corrections is presented for precision assembly. Here, an operator supervises the robot behaviours and provides corrective actions to the robot by engaging with the SpaceMouse during the policy training. The intervention can occur at any time step to optimise the policy when the robot is exploring toward the target. During an intervention, the operator can take control of the robot for up to N steps, and the intervention data are stored in both demo and RL buffers. Then, the learner picks up samples from the buffers, and adopts an actor-critic policy search method for policy parameter update [15], which is a temporal difference (TD) version of policy gradient. Briefly, the actor focuses on "choosing better actions," while the critic ensures that the feedback for those actions is accurate. This synergy drives the learning process.

Specifically, the actor, parameterised by θ_a, generates the action a_t given the current state s_t, which is achieved by sampling from the policy $\pi(\boldsymbol{a}|\boldsymbol{s};\theta_a)$, and the reward R_t and the next state $\boldsymbol{s}_{t+1}$ are returned by the action execution in the environment. The critic, parameterised by θ_c, evaluates the current policy by computing the value function $V(\boldsymbol{s};\theta_v)$, and updates the critic parameter θ_c by minimising the TD error that reflects how good the action was. Then, the policy parameters θ_a are updated in the direction suggested by the TD errors (difference between the reward and current and next states' values), which ensures actions with better-than-expected outcomes to be reinforced. This process is iterated until convergence or max episodes to generate an optimal policy.

3. Impedance-based robot control for precision assembly

3.1. Contact-rich manipulation tasks in assembly

Assembly often includes contact-rich manipulation tasks (i.e., insertion and screwing tasks) as shown in Fig. 4(a) (top), where the robot is controlled to insert a component into the hole with a high-precision tolerance, and its success relies on consistent and precise interactions between the robot and objects/the environment. This task is challenging because applying excessive force can cause the object to tilt within the gripper, resulting in failure, while insufficient force may prevent proper insertion.

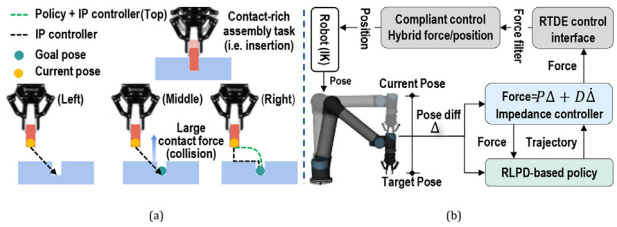

Fig. 4. (a) Contact-rich manipulation task (i.e., insertion); (b) Impedance control for contact-rich assembly tasks.

Based on the knowledge of the target pose of the robot's end-effector, an impedance controller generates part of the motion trajectories based on the pose difference (Fig. 4(a) (left)), and then the force feedback controller alters the trajectory according to the perceived contact force. However, as shown in Fig. 4(a) (middle), the trajectory in most cases would just attempt to penetrate the target object or the environment and cause the collision, since the knowledge of its geometry is not assumed. Therefore, an impedance controller-based scheme alone would not be sufficient to solve the task without causing any collision with the object. To achieve the desired behaviours/trajectories as shown in Fig. 4(a) (right), a control scheme incorporating RL policies and impedance control is adopted to manage contact-rich assembly tasks.

3.2. Constraint-based impedance controller

Impedance control regulates the relationship between force and motion. The robot's motion responds compliantly to external forces, making it ideal for tasks (i.e., contact-based operations). To ensure the success of the contact-rich assembly tasks, an impedance controller as an external controller for robots (i.e., UR5) that does not have a native one is presented for robot control. The impedance dynamics without feedforward terms is as:

$$F = k_p e + k_d \dot{e};\ e = p_c - p_t \tag{3}$$

where e is the difference between the current and target poses (p_c, p_t), and $\dot{e}$ is its differential. k_p and k_d are stiffness and damping coefficients, respectively.

Fig. 4 (b) shows the scheme of impedance control for contact-rich assembly tasks. It calculates the force and torque based on the pose difference. However, large F yielding from a big e can cause damage or hard collision when in contact with the object. Therefore, a bound b is applied on e such that $|e| \leq b$, and then $F = k_p b + 2k_d bf$ in a discrete space, where f is the control frequency. Specifically, the parameter values of k_p, k_d, b, and f used in this study are 100, 22, 0.05 (m), and 10 Hz, respectively. Since the force from the robot controller fluctuates heavily, a moving average (M) strategy is applied to form a force low-pass filter for smoothness. Specifically, it returns a mean of M=5 force signals. The calculated force and torque are sent to the RL policy for guiding the trajectory, and also to the compliant control model via the RTDE control interface. The compliant control employs an RTDE-based low-level control loop with appropriate damping to generate the robot behaviours, where the active force mode of the robot adopts a hybrid force/position control to succeed in the assembly task while maintaining safe interactions and contact.

3.3. Policy training for precise and contact-rich assembly tasks

The experiments of the policy training for contact-rich and precise assembly tasks are performed where the precision and repeatability requirements meet ISO 2768-based standard mechanical constraints. They are 1) inserting a RAM card into a slot, 2) assembling a heatsink on the motherboard; 3) assembling a fan on the heatsink to form a CPU cooler as shown in Fig. 5. A limited number of human demonstrations for task execution is crucial to accelerating the RL process. For each task, 30 trajectories of the human demonstrations by tele-operating the robot to perform tasks with the SpaceMouse are recorded. Specifically, it is defined as a success demo if the absolute difference between the current and target poses is <0.1 mm and the classification accuracy reaches over 97 % evaluated by the pre-trained reward classifier, and they work as offline demos for training initialisation.

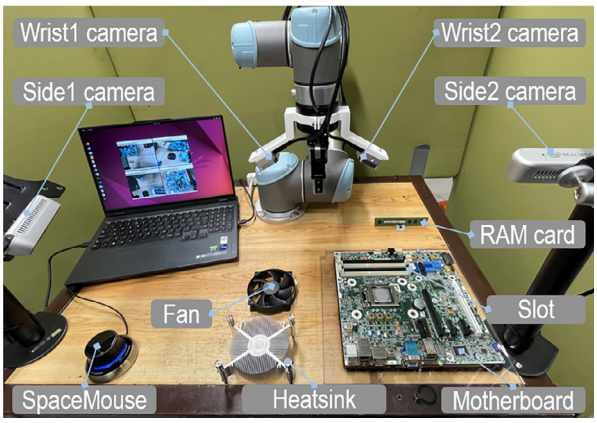

Fig. 5. Experimental setup and contact-rich assembly tasks.

Then, the policy training starts asynchronously via an actor and a learner as shown in Fig. 3. During the policy training, the human intervention provides corrective actions to the robot as necessary, until the policy is converged or the max steps are reached. The parameters used for policy training are state observation (images of two wrist cameras and robot proprioceptive states of UR5); actions (6D Cartesian pose), a ResNet-based encoder, and an Adam optimiser with a learning rate of $3e^{-5}$ and 1 million steps.

To evaluate the performance of the proposed RL-based methods for precision assembly, a set of algorithms including behaviour cloning (BC), soft actor-critic (SAC), diffusion policy (DP), and HG-Dagger [14] are selected, and they are used to compare the training performance over these tasks and perform ablation studies with the same number of human

demonstrations but different interventions. Specifically, BC, SAC and DP are trained with 100 human demonstrations, while HG-Dagger has the same number of interventions as RL. The training is done with an Ubuntu-based computer with an NVIDIA 4090 GPU in a Gym environment and running RTDE control scripts for real-time data exchange. Fig. 6 shows the success rate of the selected algorithms over these tasks. RL has shown to achieve a near-perfect success rate over all tasks, with human interventions and the support of impedance control, compared with the fluctuating success rates of other methods. The results demonstrate the superior performance of robotic RL policies (our method) for precision assembly tasks.

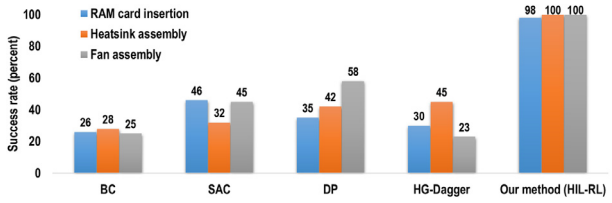

Fig. 6. Comparison of selected algorithms of RL for three assembly tasks.

The training time and cycle time (the robot completing one full execution of a task) for these tasks during the policy training are given in Table 1. The results show the sample-efficient strategy of the RL and interventions enables the policy training with a practical time, which is limited to 1.1 h even for complex assembly tasks (i.e., RAM card insertion). The cycle time over three tasks is greatly reduced, especially compared with that of the BC.

Table 1
Training and cycle time of RL policy training over various tasks

Task	SAM card insertion	Heatsink assembly	Fan assembly
Training time (h)	1.1	0.9	0.6
Cycle time (s)	5.2	4.7	3.8

4. System implementation and evaluation

The performance of the developed system and RL policies is evaluated by a precision assembly of semiconductor components at a room temperature, including the three subtasks in Section 3.3. Fig. 5 shows the experiment setup equipped with two wrist and two side cameras for visual perception and a SpaceMouse for teleoperation. The evaluation is done with 20 test cases of the whole assembly tasks, where the RTDE-based motion scripts to chain the subtasks.

The assembly process shown in Fig. 7 includes six control steps supported by the RL policies and impedance control. The assembly starts with performing the trained policy for the RAM card insertion into the slot as shown in Step ①, and the motion script controls the robot to grasp the RAM card (inset). The policy regulates the robot trajectory to first align the RAM card with the slot, which takes images from wrist cameras to detect if the alignment succeeds or not. In Step ②, appropriate force continuously calculated with pose differences is applied to execute a fine downward motion, until detecting force feedback and positive visual observations for the insertion (inset). Then, it proceeds with heatsink assembly by loading the trained policy for fine manipulation by executing the trajectories in Step ③, and aligning four pillars with holes poses a big challenge, compared with simple/single-peg-in-hole assembly, since it needs precise force-position control over robot poses. Therefore, the combination of visual observations of the pillars from wrist cameras and hybrid force-position control with a high-frequency RTDE-based protocol

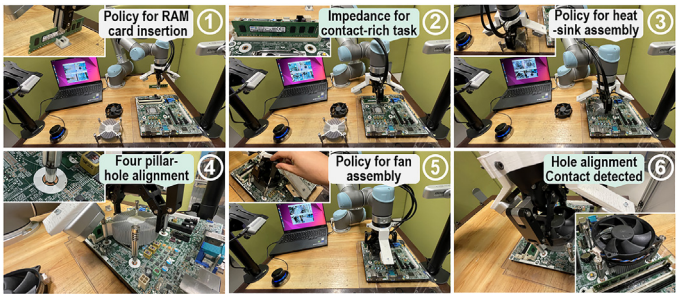

Fig. 7. Assembly process and control steps.

guides the robot for successful manipulation in Step ④, where the inset shows the result of pillar-hole insertion, followed by securing them on the motherboard. Similarly, Step ⑤ controls the robot to assemble the fan on the heatsink via calling the policy, where four small holes of the fan are aligned with those of the heatsink (in inset), through adjusting robot poses. Finally, it performs a slight downward motion with appropriate force, and ends once a contact is detected in Step ⑥. The final assembly result is shown in the inset after securing the fan on the heatsink.

Table 2
Evaluation of the system and RL policies for three assembly tasks

Task	SAM card insertion	Heatsink assembly	Fan assembly
Evaluation	18/20 (90 %)	20/20 (100 %)	20/20 (100 %)

Table 2 summarises the evaluation results of the developed system and RL policies over 20 tests, the heatsink and fan assembly reached a 100 % success rate, while two failures occurred for the RAM card insertion, which was caused by errors in gripper control. The results show good performance for assembly with RL policies.

5. Conclusions and future work

This paper presents a novel vision intelligence-based approach to precision assembly supported by RL policies and impedance control. Specific contributions of this work include:

- Learned a vision-based robotic policy with a near-perfect success rate to precision assembly, where an RLPD-based RL incorporates human demonstration corrections and a reward classifier to accelerate policy training.
- Developed an impedance controller-based hybrid force-position control for precise and closed-loop robot control to succeed execution of contact-rich manipulation tasks.
- Demonstrated a sample-efficient and robust robotic assembly policy by an agent-environment Gym and a visual encoder, enabling complex vision-based precision assembly.

This study provides an efficient scheme to train and learn vision-based robotic RL policy for contact-rich manipulation tasks in precision assembly. It serves as a general-purpose framework for policy training for acquiring various types of control strategies. Human interventions adopted in the study can significantly increase the efficiency and performance of the policy training by providing corrective actions. Meanwhile, an impedance controller is proposed to perform fine hybrid force-position manipulation control. Compared with conventional methods for precision assembly (e.g., online and offline programming methods or simulation) [2,16], our method demonstrates not only superior performance in handling dynamic and precise assembly tasks but with much less effort for environmental variation adaptation. This study serves as the basis for future learning-based robotic manipulation research. Our future work will focus on deployable and dexterous robotic manipulation skills capable of adapting to diverse environments and assembly tasks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Sichao Liu: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Lihui Wang:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Acknowledgments

The authors acknowledge support from the Vetenskapsrådet under award 2023-00493, the Berzelius-2024-124, and the NAISS 2024/5-164.

References

[1] Valente A, Baraldo S, Carpanzano E (2017) Smooth trajectory generation for industrial robots performing high precision assembly processes. *CIRP Annals* 66 (1):17–20.

[2] Luo J., Xu C., Wu J., Levine S. (2024) Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. doi:10.48550/arXiv.2410.21845.

[3] Zhang H, Solak G, Lahr GJG, Ajoudani A (2024) SRL-VIC: a variable stiffness-based safe reinforcement learning for contact-rich robotic tasks. *IEEE Robot Autom Lett* 9 (6):5631–5638.

[4] Makris S, Dietrich F, Kellens K, Hu SJ (2023) Automated assembly of non-rigid objects. *CIRP Annals* 72(2):513–539.

[5] Tang C, Abbatematteo B, Hu J, Chandra R, Martín-Martín R, Stone P (2024) Deep reinforcement learning for robotics: a survey of real-world successes. *Annual Review of Control, Robotics, and Autonomous Systems* 8:1–48.

[6] Kober J, Bagnell JA, Peters J (2013) Reinforcement learning in robotics: a survey. *Int J Rob Res* 32(11):1238–1274.

[7] Li C, Zheng P, Yin Y, Wang B, Wang L (2023) Deep reinforcement learning in smart manufacturing: a review and prospects. *CIRP Journal of Manufacturing Science and Technology* 40:75–101.

[8] Ankile L., Simeonov A., Shenfeld I., Torne M., Agrawal P. (2024) From imitation to refinement – residual RL for precise assembly. doi:10.48550/arXiv.2407.16677.

[9] Song Y, Romero A, Müller M, Koltun V, Scaramuzza D (2023) Reaching the limit in autonomous racing: optimal control versus reinforcement learning. *Sci Robot* 8 (82):eadg1462.

[10] Ball PJ, Smith L, Kostrikov I, Levine S (2023) Efficient online reinforcement learning with offline data. In: *Proceedings of the 40th International Conference on Machine Learning*, 1577–1594.

[11] Towers M., Kwiatkowski A., Terry J., Balis J.U., De Cola G., Deleu T., Goulao M., Kallinteris A., Krimmel M., KG A., Perez-Vicente R. (2024) Gymnasium: a standard interface for reinforcement learning environments. doi:10.48550/arXiv.2407.17032.

[12] Bongratz F., Golkov V., Mautner L., Libera L.D., Heetmeyer F., Czaja F., Rodemann J., Cremers D. (2024) How to choose a reinforcement-learning algorithm. doi:10.48550/arXiv.2407.20917

[13] Horgan D., Quan J., Budden D., Barth-Maron G., Hessel M., van Hasselt H., Silver D. (2018) Distributed prioritized experience replay. doi:10.48550/arXiv.1803.00933.

[14] Kelly M, Sidrane C, Driggs-Campbell K, Kochenderfer MJ (2019) HG-DAgger: interactive imitation learning with human experts. *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, Montreal, QC, Canada, 8077–8083.

[15] Stamer F, Lanza G (2023) Dynamic pricing of product and delivery time in multi-variant production using an actor critic reinforcement learning. *CIRP Annals* 72 (1):405–408.

[16] Inoue T, De Magistris G, Munawar A, Yokoya T, Tachibana R (2017) Deep reinforcement learning for high precision assembly tasks. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Vancouver, BC, 819–825.