Journal of Intelligent & Robotic Systems (2023) 109:57
https://doi.org/10.1007/s10846-023-01970-8

REGULAR PAPER

Reinforcement Learning With Stereo-View Observation for Robust Electronic Component Robotic Insertion

Grzegorz Bartyzel[1] · Wojciech Półchłopek[2] · Dominik Rzepka[2]

Received: 19 March 2023 / Accepted: 15 September 2023 / Published online: 31 October 2023

Abstract
In modern manufacturing, assembly tasks are a major challenge for robotics. In the manufacturing industry, a wide range of insertion tasks can be found, from peg-in-hole insertion to electronic parts assembly. Robotic stations designed for this problem often use conventional hybrid force-position control to perform preprogrammed trajectories, such as e.g. a spiral path. However, electronic parts require more sophisticated techniques due to their complex geometry and susceptibility to damage. Production line assembly tasks require high robustness to initial position and rotation variations due to component grip imperfections. Robustness to partially obscured camera view is also mandatory due to multi stage assembly process. We propose a stereo-view method based on reinforcement learning (RL) for the robust assembly of electronic parts. Applicability of our method to real-world production lines is verified through test scenarios. Our approach is the most robust to applied perturbations of all tested methods and can potentially be transferred to environments unseen during learning.

Keywords Reinforcement learning · Assembly · Industrial robotics · Electronic parts insertion · Stereo-view observation

1 Introduction

1.1 Background

Despite the progress in the robotization of the industry, there are still many assembly tasks that are usually performed manually by production workers. The need for further improvements in production efficiency and cost reduction has inspired research for many years. Most of the work has focused on an idealised assembly model, known as the peg-in-hole problem [1]. However, due to the diversity of assembled components' shapes in the manufacturing industry, this is a subject of ongoing research [2].

✉ Grzegorz Bartyzel
gbartyzel@agh.edu.pl

Wojciech Półchłopek
wojciech.polchlopek@fitech.pl

Dominik Rzepka
dominik.rzepka@fitech.pl

[1] Department of Automatic Control and Robotics, AGH University of Science and Technology, al. Adam Mickiewicza 30, Kraków 30-059, Poland

[2] Fitech, ul. Kościelna 5, Sucha Beskidzka 34-200, Poland

In the electronics manufacturing industry, industrial robots face challenges when assembling non-standardised electronic components in through-hole technology (THT). This difficulty comes from the physical properties of these components, which come in various shapes and have differing numbers of leads arranged in non-standardised patterns. The pins are also easily bent due to their susceptibility to applied forces. Furthermore, the clearance of through-hole pins is typically less than 1 mm, depending on the printed circuit board (PCB) design, making inserting electronic parts challenging. Figure 1 presents a close view of the THT component and the effect of damaged pins on the element.

A highly precise force control system is necessary to mitigate potential damages. However, industrial assembly systems must also account for errors introduced by the grasping procedure and the PCB clamping mechanism. The picking error arises from the way electronic parts are fed to the robotic stations through profiled trays with large clearance slots.

1.2 Related Work

Robotic stations used for assembly on production lines are based mainly on compliance control systems. These stations use impedance or admittance controllers to control industrial

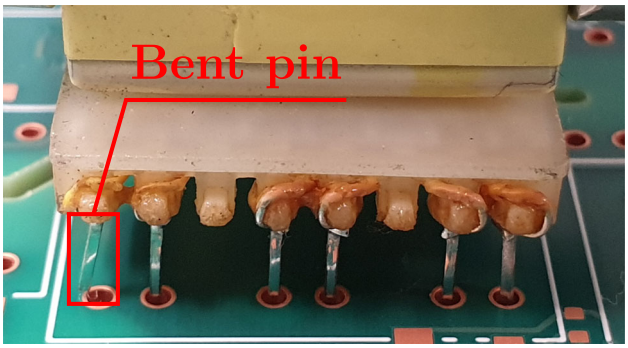


Fig. 1 A close view of the THT component with one of the pin bent

robots [3, 4] that perform programmed trajectories while maintaining a constant downforce. Nevertheless, these control systems require manual parameter adjustment, which is time-consuming.

The methods based on the compliance control system require high sensor precision and well-prepared robotic stations. However, machine vision and deep learning can improve the performance of insertion tasks, even when dealing with imperfect hardware or the construction of the robotic station. For example, Huang et al. [5] propose a pure machine vision system with a feedback rate of 1000 Hz to align a peg in a hole instead of a force control system. Meanwhile, deep learning-based methods for insertion tasks are presented by Triyonoputro et al. [6] and Yu et al. [7]. Both solutions use the convolutional neural network (CNN) to precisely compute the pose of the insertion target. Moreover, Triyonoputro et al. [6] use two images captured by the vision tool attached to the robot end effector. The trajectory algorithm then uses this computed pose as input.

Recently, reinforcement learning (RL) has gained attention as a solution to assembly problems [8–11]. These works utilise RL algorithms for much more complex assembly tasks than peg-in-hole insertion, such as connector or gear insertion. The RL agent commands the robot controlled by an impedance or an admittance control system in all these methods. The agent's observation mainly consists of proprioception and 6-axis force-torque (F/T) data. Another group contains RL methods that rely solely on visual information. In Schoettler et al. [11] work, the agent acquires the image from an external camera placed in the workspace. In [12, 13], visual information is captured from a single wrist camera and preprocessed by a complex neural network pre-trained in a supervised manner.

The aforementioned works have addressed the solution for peg-in-hole or connector insertion tasks. However, assembly of electronic parts is considered a multiple peg-in-hole task, which is more challenging for industrial robots due to the complex geometry of the object [14, 15]. To tackle this issue, Hou et al. [16] have proposed an RL-based method that employs a DDPG algorithm [17] supported by a fuzzy logic system and variable time-scale prediction. In this method, the RL agent computes the 6-dimensional action, which represents translations and rotations along the XYZ-axes based on the object's pose relative to the target and the 6-axis F/T data. Similar approaches have been introduced by Hou et al. [18] and Xu et al. [19]. In both works, DDPG is a core algorithm, and the policy network's output is used to correct the control signal computed by the manually tuned PD force controller. However, these works differ in the reward function they use. Xu et al. [19] propose a fuzzy reward system instead of a complex handcrafted reward function. The improvements proposed in these works are intended to accelerate training and achieve safe exploration. However, the manipulated objects used in those works were solid metal blocks that were more resistant to damage than electronic components. Additionally, in the presented experiments, those objects were rigidly attached to the robot's end-effector, simplifying the problem by reducing the impact of uncertainties from the grasping procedure.

In contrast, Ma et al. [20] propose a reinforcement learning-based solution for the assembly of electronic components, such as the pin header. Their solution uses high-quality cameras and a precise 6-axis F/T sensor. Nevertheless, the agent's observation space consists of only F/T data, and cameras are used for the pre-policy control step. The action space compared to the abovementioned works computes only translations along XYZ-axes.

1.3 Contribution

This paper presents a method for the precise assembly of non-standardised THT electronic components using reinforcement learning and stereo-view observation. We employ Soft Actor-Critic (SAC) [21] as the core algorithm, as it has been shown [22] to be more efficient for real-world robotics applications than other continuous control algorithms such as PPO [23] and TD3 [24]. We refer to our method as *SAC with stereo-view observation space* (SAC-SV). Furthermore, we used two separate convolutional neural networks to extract features from input images, as opposed to the single network used in previous work [6].

Moreover, this work provides test scenarios that are suitable for evaluating the potential of a method to be used in a real-world production line. We collected the requirements specified by production personnel and identified potential sources of errors that could occur in robotic stations. Our test scenarios assess the robustness of the methods to position and rotation disturbances. Furthermore, the proposed procedures verify whether the trained agent can be transferred from experimental to production scenarios. Specifically, for

the electronic parts assembly task, we check if the policy trained on the empty PCB can be applied to the PCB after the automatic assembly stages.

The paper is organised as follows. Section 2 introduces a method for assembling THT electronic parts. We start by describing the environment for the electronic component assembly task. Next, we present the industrial robot control system and the technique for asynchronous learning. Section 3 presents the experiments along with a detailed description of the robotic system used for the experiments and the training procedure. In this section, we compare SAC-SV to state-of-the-art approaches for vision-driven reinforcement learning driven by a single camera vision system acquired from an external camera or the tool-mounted camera. We also validate our solutions against conventional methods and the force-based RL method. Following these experiments, we report the performance of the proposed method in transferring the policy trained on the empty PCB to a scenario with a partially assembled PCB. Finally, we discuss the results obtained and plans for future work.

2 Method

2.1 Reinforcement Learning

We model our problem as a standard RL setting [25], where interaction between the agent and the environment can be described as a Markov decision process (MDP). In each discrete timestamp t, the agent is in state $\mathbf{s}_t \in \mathcal{S}$, performs the action $\mathbf{a}_t \in \mathcal{A}$, and receives the scalar reward r_t sampled from the reward function $r_t(\mathbf{s}_t, \mathbf{a}_t)$, where $\mathcal{S}$ defines the state space of the environment, and $\mathcal{A}$ defines the continuous action space. After performing an action $\mathbf{a}_t$, the environment moves to the next state, which is drawn at random from an unknown state transition distribution $\mathbf{s}_{t+1} \sim p(\cdot \mid \mathbf{s}_t, \mathbf{a}_t)$. The objective of the RL agent is to learn the policy $\mathbf{a}_t = \pi(\mathbf{s}_t)$ from the collected data by maximising the expected return $R = \sum_{t=0}^{T} \gamma^t r_t$, where T is the length of the planned trajectory and $\gamma \in (0, 1)$ is a discount factor.

2.2 Soft Actor-Critic

The Soft Actor-Critic [21] is an actor-critic off-policy algorithm based on maximal entropy. Entropy controls the agent's exploration ability by augmenting the reward at each step. SAC uses neural networks as approximations for soft Q-function $Q_\psi(\mathbf{s}_t, \mathbf{a}_t)$ and policy $\pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t)$, which are parameterized respectively by ψ and ϕ. The soft Q-function parameters can be updated by minimising the *soft Bellman residual*

$$J_Q(\psi) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \sim \mathcal{D}} \Bigg[\frac{1}{2} \big(Q_\psi(\mathbf{s}_t, \mathbf{a}_t) - (r(\mathbf{s}_t, \mathbf{a}_t) + V_{Q_{\psi_1, \psi_2}}(\mathbf{s}_{t+1}) \big)^2 \Bigg], \tag{1}$$

where $\mathcal{D}$ is an experience replay buffer that stores transitions $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{s}_{t+1})$, $V_{Q_{\psi_1, \psi_2}}(\mathbf{s}_{t+1})$ denotes the value function implicitly defined by Q-functions and policy as $\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi} \Big[\min_{i \in \{1,2\}} Q_{\bar{\psi}_i}(\mathbf{s}_t, \mathbf{a}_t) - \alpha \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t) \Big]$. $Q_{\bar{\psi}}$ is a target soft Q-function, and α is an entropy temperature coefficient. The parameters $\bar{\psi}$ of the target soft Q-function are obtained as an exponential moving average of the weights of the soft Q-function. The Gaussian policy parameters ϕ are trained by minimizing

$$\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}, \mathbf{a}_t \sim \pi_\phi} \Big[\alpha \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t) - \min_{i \in \{1,2\}} Q_{\psi_i}(\mathbf{s}_t, \mathbf{a}_t) \Big] \tag{2}$$

using the reparameterization trick [26]. Finally, the entropy temperature coefficient α can be fixed during training or dynamically adjusted, as proposed by [27]. This coefficient can be optimised by solving the following objective:

$$J(\alpha) = \mathbb{E}_{\mathbf{a}_t \sim \mathcal{D}} \big[\alpha \log \pi_\phi(\mathbf{a}_t \mid \mathbf{s}_t) - \alpha \bar{\mathcal{H}} \big] \tag{3}$$

where $\bar{\mathcal{H}}$ is a target entropy that is usually set empirically to $\bar{\mathcal{H}} = \dim(\mathcal{A})$. We follow the implementation proposed by Haaranoja et al. [27], where two soft Q functions with independent parameters Q_i are used to mitigate positive bias in the policy improvement steps. The target Q-value is computed by taking the minimum value from the Q-function approximations. Both networks are independently optimised by solving the $J_{Q_i}(\psi_i)$ objectives.

2.3 Assembly Process Environment

In our environment for assembling electronic parts tasks, the RL agent's observation contains two images acquired from cameras attached to the end-effector. Figure 2 illustrates the concept of obtaining these images from this vision tool. The camera's view angle was empirically chosen to achieve a view that gives information about the assembly place and its surroundings. Each output image is of size 1024×512 pixels in the RGB colour space. However, to enable the use of the neural network in a real-time control scenario, the desired images are resized to the resolution of 128 × 128 pixels.

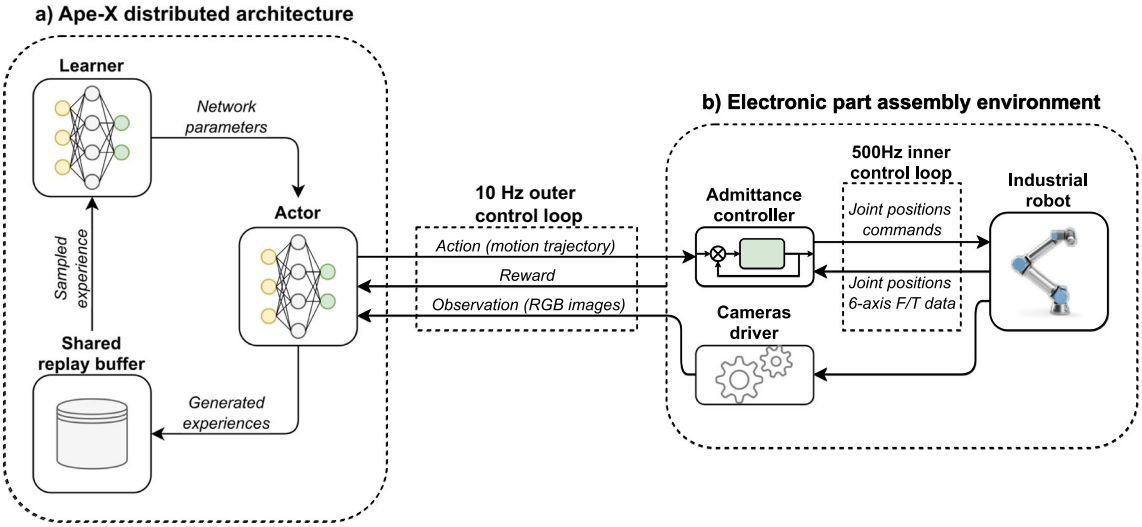


Fig. 2 Images acquisition visualisation from the vision tool

Due to the specificity of the task and direct control of the real robot, we have implemented constraints for the RL agent. A workspace is defined as a cylinder with a radius of 7 mm and a limitless height. The insertion pose of electronic parts determines the reference frame of the workspace. Additionally, the assumed maximum rotation about each axis is 15°. At the start of each episode, the agent is positioned 2 mm above the PCB surface.

Each trial lasts up to 50 steps. During a single step, the agent executes the action $\mathbf{a}_t = [\Delta x_t, \Delta y_t, \Delta z_t, \theta_t^z]$, where Δx_t, Δy_t, Δz_t are displacements along the XYZ-axes, respectively, and θ_t^z is a rotation on the Z-axis. Thus, the action space is 4-dimensional. The range of action space is $[-1.0,\ 1.0]$ mm for displacements and $[-0.5°,\ 0.5°]$ for rotations. For each non-terminal step, the agent receives a reward

$$r = -\tanh(\alpha \cdot d) \tag{4}$$

where d is a ℓ_2-distance between the tool center point (TCP) and full insertion pose, and α is a reward sensitivity coefficient. Full insertion occurs when the robot reaches the assembly position in the XY-axes and the defined position in the Z-axis below the surface of the PCB. To ensure safety, we designed a penalisation mechanism in the environment. The episode is interrupted when the agent leaves the workspace, exceeds the time limit, or exceeds the rotation limit. If the episode is terminated due to leaving the workspace or exceeding the time limit, the agent receives the same reward as during nonterminal steps. If termination occurs due to exceeding the rotation limit, the agent receives a reward of $r = -2$ to prevent damage to camera cables connected to the vision tool. The task is completed when the relative position of the TCP p^z on the Z-axis is less than or equal to 0.0 mm, which means that the electronic part is inserted into the target position. In this situation, the reward received is $r = 10.0$.

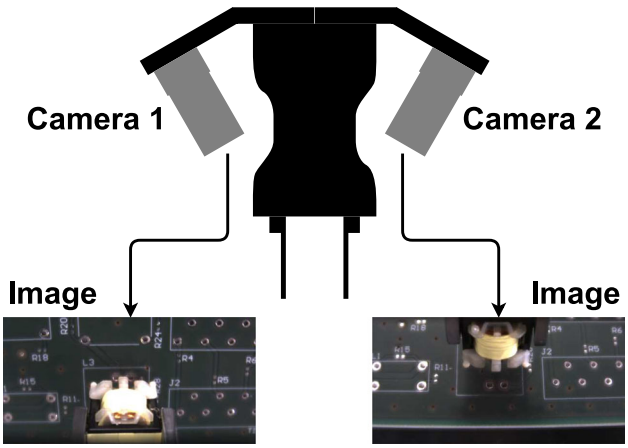


Fig. 3 The block diagram of the RL-based method for assembling electronic parts: a) Ape-X distributed architecture that allows asynchronous training; b) the environment for assembling electronic parts with presented control diagrams. In this setup Actor (the RL agent) sends the Cartesian motion trajectory to the admittance controller, which directly controls the robot's joints

2.4 Smart Assembling

We developed an RL-based method for assembling electronic parts, integrating the SAC algorithm with the admittance control system. The presented system consists of two control loops. The block diagram is presented in Fig. 3. In the outer loop, the controlling element is the SAC algorithm, and in the

inner control loop, the admittance control system [28] is used. The RL agent sends commands to the admittance controller at 10 Hz, receiving feedback information at the same frequency. The pose information required for the reward function is received from the admittance controller. At the same time, the images are acquired from the camera's drivers running independently from the controller. To ensure the reliable real-time control loop that sends commands with a given frequency, we integrated the SAC algorithm with a distributed learning architecture called Ape-X [29]. In this architecture, multiple actors are spawned, each with its instance of the environment. Those actors generate the experiences and store them in the shared replay buffer. The learner samples mini-batches from this shared replay buffer and updates the network parameters. The actors' parameters are periodically synchronised with the latest learner's parameters. Our experiments were conducted with only one robot, so we used Ape-X architecture with one spawned actor.

2.5 Policy Model

We represented the control policy as a neural network, as introduced in Section 2.2. As described in Section 2.3, the observation space consists of RGB images. Each image is pre-processed by an independent 5-layer convolution neural network (CNN) with filters of size 32. Then, the computed features are concatenated and passed directly to the actor π_ϕ and the critic Q_ψ. Both function approximators are neural networks with two fully connected layers of 256 neurones per layer size. For every layer in the model, we use LeakyReLU [30] as an activation function. The concept diagram of the model is depicted in Fig. 4

The CNN backbones are shared between the actor and critic networks in variants with visual information. We followed an optimisation procedure proposed by Yarats et al. [31], where the parameters of the vision network are updated by the gradient calculated from the critic loss function.

2.6 Admittance Controller

We implemented a standard admittance controller [28] operating in the task space to safely assemble the electronic parts susceptible to applied forces. This controller is part of the control scheme depicted in Fig. 3. Compared to hybrid force-position control, this control system allowed us to control the robot with high precision in the task space and minimise the contact force detected during trajectory execution. The admittance controller is described by

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{D}\dot{\mathbf{x}}(t) + \mathbf{K}(\mathbf{x}(t) - \mathbf{x_d}) = \mathcal{W}^{ext}(t) \tag{5}$$

where $\mathbf{K}$, $\mathbf{D}$, and $\mathbf{M}$ represent stiffness, damping, and inertia matrices, respectively. $\mathcal{W}^{ext} = [F, \tau]$ represents the

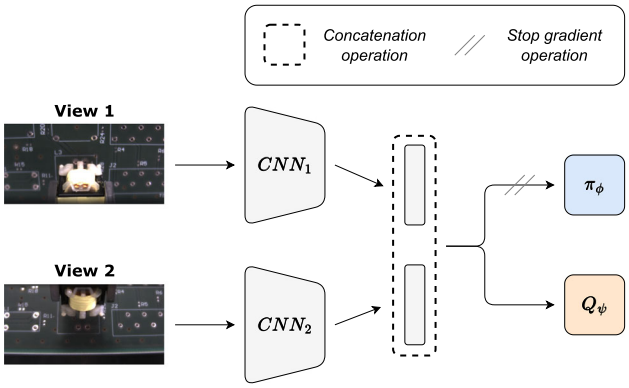


Fig. 4 Soft Actor-Critic with stereo-view observation (SAC-SV): architecture built with two separate convolution neural networks for each view and two fully connected neural networks, respectively, for actor and critic. Features computed by vision networks are concatenated and passed directly to the actor and critic

contact forces and torques, $\mathbf{x_d}$ is the motion of the target end-effector, and $\mathbf{x} = [p, \theta]$ is the control output, which represents the pose of the robot's fingertip. The coefficients d_{ij} of the damping matrix can be calculated using the formula $d_{ij} = 2\zeta_{ij}\sqrt{m_{ij}k_{ij}}$, where ζ_{ij} is a damping ratio for each degree of the control system. The control signal is first computed by integrating the acceleration $\ddot{\mathbf{x}}$ and the obtained velocity $\dot{\mathbf{x}}$. The control acceleration $\ddot{\mathbf{x}}$ is given by

$$\ddot{\mathbf{x}}(t) = \mathbf{M}^{-1}(\mathcal{W}^{ext}(t) - \mathbf{D}\dot{\mathbf{x}}(t) - \mathbf{K}(\mathbf{x}(t) - \mathbf{x_d})) \tag{6}$$

Finally, the robot's joint positions $\mathbf{q}(t)$ are computed from inverse kinematics applied to the resulting control output.

3 Experiments and Results

3.1 Experimental Setup

We built a real-world laboratory stand to carry out experiments, depicted in Fig. 5. This laboratory stand consists of the following devices: the Universal Robot UR5e-series industrial robot[1], the servo-electric gripper[2], a 6-axis F/T sensor[3], and a custom-made tool with vision sensors. We placed PCB panels and electronic components in the robot's workspace. In the production lines, PCBs are delivered in panels, where a single panel can contain different numbers of boards. The electronic parts are placed on the 3D printed trays. Such a setup allowed us to ensure conditions similar to those on the production line.

[1] https://www.universal-robots.com/products/ur5-robot/

[2] https://robotiq.com/products/hand-e-adaptive-robot-gripper

[3] https://www.ati-ia.com/products/ft/ft_models.aspx?id=Axia80-M20&campaign=axia80

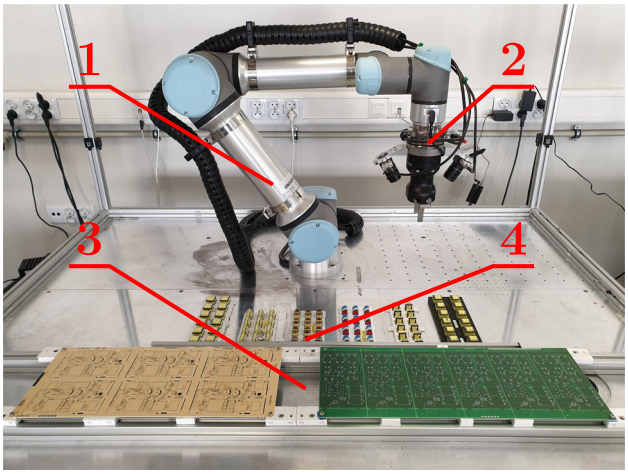


Fig. 5 The laboratory stand used for experiments. It consists of 1 - industrial robot Universal Robot UR5e-series; 2 - a tool with gripper, F/T sensor and cameras; 3 - various PCBs panels used for the experiments; 4 - trays with electronic parts

In our experiments, we used two distinct PCB panels, one designed for research purposes and one sourced from the production line. We selected three types of electronic parts, namely: component type 1a/b (Fig. 6a and b), component type 2 (Fig. 6c), and component type 3 (Fig. 6d). The letters *a* and *b* denote the various types of PCB for specific electronic parts. These elements differ in their geometry, appearance, and arrangement of the leads. Figure 6 presents the electronic parts and their corresponding insertion places.

We used ROS 2 [32] middleware to control the industrial robot and operate peripheral devices. Furthermore, we used RLlib [33] as software to manage the learning process and implement the RL agents. The advantage of RLlib is the availability of out-of-the-box software for distributed algorithms like Ape-X. We performed all experiments on the workstation with NVIDIA Titan X GPU[4].

3.2 Training and Evaluation

During the training process, the agent's task was to insert the electronic component into the target pose on the PCB. The agent was trained for 50000 steps in an asynchronous manner, as described in Section 2.4, which took an average of 3 hours. In this setup, the actor sends a rollout with 10 transitions to the replay buffer while the learner synchronises model parameters every 50 environment steps. Each episode began with picking an electronic element from the tray. Followed by the robot moved to the initial pose, which is the electronic part's assembly pose 2 mm above the PCB surface. Moreover, to ensure that each episode was unique and improve the robustness of the RL agent, the initial position on the XY-axes and the initial rotation on the Z-axis were disturbed with the noise sampled, respectively, from $p^{xy}_{noise} \sim \mathcal{U}(-2, 2)$ mm and $\theta^{z}_{noise} \sim \mathcal{U}(-2°, 2°)$. After the termination of the episode, the robot would return the grasped electronic part and pick up another one.

Next, we evaluated each trained model with respect to its robustness to environmental disturbance. We designed a test scenario that reflects the cumulative errors in the production machinery. There are three primary sources of errors in robotic assembly system on the production line: determining the picking pose of the electronic parts placed in the trays, a picking procedure with universal fingers, and a panel clamping system precision.

The test scenario consisted of 7 tests. These tests differed in the continuous uniform distribution range applied to the XY-axes' initial position and the Z-axis's initial rotation. At first, the model was evaluated without any applied noise. Afterwards, the robustness of the model against position disturbance was tested by applying noise samples from $p^{xy}_{noise} \sim \mathcal{U}(-2, 2)$ mm and $p^{xy}_{noise} \sim \mathcal{U}(-5, 5)$ mm. Subsequently, we performed the evaluations against the rotation disturbance sampled from $\theta^{z}_{noise} \sim \mathcal{U}(-3°, 3°)$ and $\theta^{z}_{noise} \sim \mathcal{U}(-7°, 7°)$. Finally, the model was subjected to tests with compound perturbations. For each test, we run 100 trials of insertion. During the evaluation, we collected data on the insertion status (success or failure) and assembly time of the successfully completed trial. On the basis of these data, we calculated the success rate and averaged the assembly time for each test.

For every performed experiment, we set the admittance controller's desired stiffness and inertia as the following diagonal matrices:
$\mathbf{K} = \text{diag}\{1000, 1000, 1000, 20, 20, 20\}$,
$\mathbf{M} = \text{diag}\{3, 3, 3, 0.04, 0.04, 0.04\}$.
and a one-damping ratio $\zeta = 2.8$ for every degree of freedom. To ensure smooth motion and stability of the control system, we limited linear velocities to 0.1 m/s and angular velocities to 1.0 rad/s. Moreover, we used a low-pass filter with a cut-off frequency of 25 Hz for data acquired from the F/T sensor and applied the following constraints: 5 N for forces and 1 Nm for torques. We use the default joint-torque limits provided by the vendor[5]. The parameter values were empirically determined to maximise movement speed while maintaining compliance. Further details on the SAC algorithm hyperparameters are available in Table 1.

[4] https://www.nvidia.com/en-us/geforce/products/10series/titan-x-pascal/

[5] https://www.universal-robots.com/articles/ur/robot-care-maintenance/max-joint-torques/

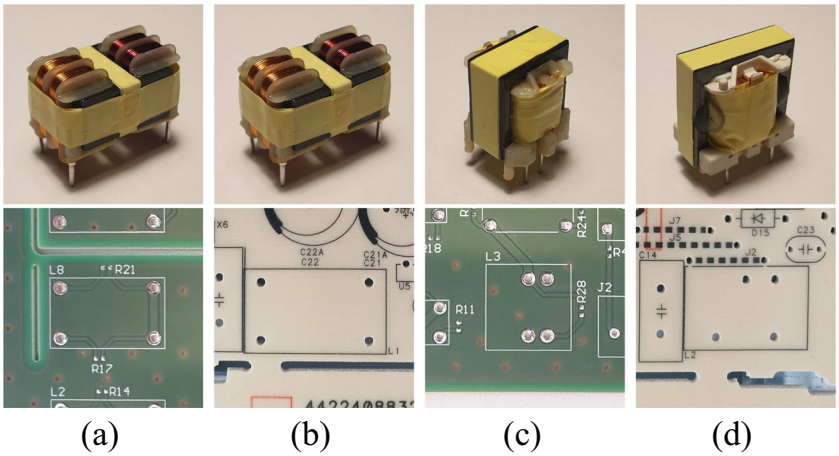


Fig. 6 Electronic parts (above) and their corresponding insertion positions (below) used for the experiments. This work proposes the following naming convention for electronic parts: a) component type 1a, b) component type 1b, c) component type 2 and d) component type 3. All shown components have four leads

3.3 Performance Comparison

In these experiments, we evaluate the suitability of our dual-camera robotic vision system for assembling non-standardised electronic parts by comparing the method to the other vision-driven RL methods with different visual information sources and conventional methods, which are widely used in the industry. The following methods and setups are compared:

1) **Straight down** - The robot moves straight down from the starting position until successful insertion or exceeding the limit of contact force, which we set to 2 N.
2) **Random search** - The robot moves randomly in the XY plane until the insertion or trial termination signal is detected. The displacements on those axes are sampled from the uniform distribution. The force controller controls the displacement in the Z-axis by holding a constant contact force of 2 N.
3) **Spiral search** - The robot follows the spiral trajectory [34] on the XY plane until successful insertion or a trial termination signal occurs. The force controller controls the displacement in the Z-axis by holding a constant contact force of 2 N.
4) **SAC with the combined view (SAC-CV)** - SAC-CV, like our SAC-SV, learns the policy that takes multiple images on the input to the neural network. However, the input images are combined into one, as was presented by Triyonoputro et al. [6]. A detailed description of this operation is given in Appendix A.1.
5) **SAC with the mono view (SAC-MV)** - SAC-MV uses an image acquired from a single camera vision system attached to the robot's end-effector like it was presented in [12]. However, in our experiments, we used SAC instead of DDPG, which was initially introduced in the work mentioned.
6) **SAC with the external view (SAC-EV)** - SAC-EV differs from the previous in the source of visual information. Images for the action computation are acquired from an external camera placed in the robot workspace [11]. The testbed for this experiment is presented in Appendix A.2. However, in this setup, the camera's field of view was set so that the agent could see only one PCB from the entire panel, consisting of a group of PCBs.
7) **SAC with F/T feedback (SAC-Force)** - The SAC-Force [20] differs from previous methods by taking 6-axis F/T data $[F_x, F_y, F_z, M_x, M_y, M_z]$ to compute the output action. Input F/T data were acquired by averaging 24 received samples. Here, we use the same actor and critic neural networks as vision-based agents.

Table 1 SAC algorithm hyperparameters used for experiments

Hyperparameters	Value
Replay buffer size	50000
Batch size	256
Discount γ	0.95
Optimizer	Adam
Actor learning rate	0.0003
Critic learning rate	0.0003
Temperature learning rate	0.0003
Initial temperature	0.1
Critic soft-update rate τ	0.005

In this section, we present results only for component type 1a (Fig. 6a), while in Appendix B, we show results for the remaining electronic parts. We followed all the evaluation procedures for all the experiments described in Section 3.2. The methods mentioned were evaluated based on the success

rate and the average assembly time. The results obtained for component type 1a are reported in Table 2.

Due to the approach to image acquisition by SAC-EC, we have only provided the results of the tests performed on the PCB used for training. During the experiments, we also evaluated the effectiveness of this method on other PCBs from the panel. However, we omitted them in Table 2 since the agent was unable to perform the task. Additionally, we attempted to train an RL agent whose external camera returns the image of the entire panel and not a single PCB. However, the agent barely achieved more than 20% efficiency during learning. Therefore, we decided to stop further experiments with it.

The only methods that were able to achieve an almost 100% success rate were those driven by visual observation space. Moreover, our method was the most reliable among them. The RL agent with F/T feedback information reported poor performance in all test scenarios. We assume that this was caused by the fact that Ma et al. [20] did experiments for precise electronic parts assembly tasks with a more complex test bed consisting of a specialised industrial robot and high-resolution cameras. Furthermore, their experimental object was a pin header with pins aligned in a line instead of a more complex pattern.

Additionally, we examined another approach to the stereo-view observation space called SAC-CV. We modified the image processing technique presented by Triyonoputro et al. [6] to fit into the reinforcement learning domain. SAC-CV achieved comparable results to our method and even scored slightly lower average assembly times for the test scenarios with minor perturbations. However, overall, SAC-SV is more robust to the increasing applied disturbances. In terms of the production application, our solution does not need additional devices, like vision systems, that reduce the error that occurs.

The above results show that the insertion of electronic components is a challenge for conventional methods. The holes for non-standardised THT electronic parts have tight clearance, significantly complicating the assembly process. Traditional techniques perform programmed trajectories, which rely only on pose feedback information; therefore, their effectiveness decreases with increasing disturbances. Furthermore, the standard implementation of these algorithms cannot handle compound and orientation perturbations. With the feedback from the vision system, our method learns the features that enable it to be robust against all applied disturbances. In addition, off-policy algorithms like

Table 2 Summary of the experiments comparing the performance of the methods for the component type 1a (Fig. 6a)

Methods	Without noise	Position		Orientation		Mix	
		±2 mm	±5 mm	±3°	±7°	±2 mm, ±3°	±5 mm, ±7°
	Assembly success rate						
SAC-SV (ours)	**100%**	**100%**	**70%**	**100%**	**100%**	**100%**	**62%**
SAC-CV	**100%**	**100%**	60%	**100%**	98%	**100%**	53%
SAC-MV	**100%**	98%	66%	94%	87%	97%	53%
SAC-EV	**100%**	94%	48%	**100%**	85%	93%	56%
SAC-Force	60%	10%	2%	38%	18%	2%	0%
Straight down	69%	0%	0%	42%	8%	0%	0%
Random search	81%	23%	4%	46%	28%	13%	0%
Spiral search	77%	26%	4%	52%	19%	6%	0%
	Assembly time [s]						
SAC-SV (ours)	1.90±0.19	**1.79**±0.22	3.45±1.44	1.95±0.21	2.52±0.78	2.07±0.32	4.16±1.49
SAC-CV	**1.82**±0.19	1.87±0.19	3.19±1.37	**1.85**±0.18	**2.27**±0.85	**1.92**±0.25	**3.79**±1.29
SAC-MV	2.03±0.48	2.12±0.63	3.37±1.34	2.22±0.91	2.91±1.23	2.22±0.71	3.95±1.46
SAC-EV	2.07±0.28	2.20±0.62	**3.05**±1.05	2.18±0.25	2.97±1.14	2.57±0.88	4.16±1.09
SAC-Force	3.27±1.51	4.62±0.81	5.44±1.30	3.88±1.52	4.53±1.11	4.85±0.48	–
Straight down	2.44±0.96	–	–	2.46±0.82	2.62±0.89	–	–
Random search	2.35±0.74	3.11±0.92	3.16±1.04	2.63±0.78	2.59±0.62	2.92±0.80	–
Spiral search	2.89±0.94	3.21±1.10	2.09±0.18	2.55±0.70	2.63±0.79	2.38±0.62	–

The 100 trials of the insertion validated each method. Conventional methods such as spiral or random search cannot achieve high success rates due to the complexity of the task. Only vision-driven RL-based solutions achieve high success rates. Nevertheless, our method is the most robust to the applied disturbances. The bottom part of the table shows the average assembly times with standard deviations

Bold entries highlight the best results in a single column to improve visualization for the reader. In the column's top part, positions with the highest success rates are bolded. The positions with the lowest assembly times are highlighted in the bottom part

SAC decide on the next action at each time step to quickly correct the trajectory.

3.4 Transfer to Partially Assembled PCB

In the previous experiments, the agents were trained on the empty PCB. However, in real-world scenarios, the final production stage is an assembly of the non-standardised THT parts. The ideal approach would be to train the policy offline, outside the production line, and then transfer it to the robotic station at the factory. In this particular experiment, we evaluated vision-based RL agents in terms of their possible transferability to the partially assembled PCB (Fig. 7.)

Table 3 presents the obtained results. All RL agents trained from scratch with the visual feedback acquired from the vision sensors attached to the end-effector achieved an almost 100% success rate. However, the variant's performance using an external camera as the observation space source was significantly worse because of other components' occlusion of the assembly place. When evaluating the performance of the policy transfer from an empty PCB, none of the agents achieved a 100% success rate. Nevertheless, our method achieved the highest efficiency among them. This experiment showed that the stereo-view observation space can score relatively good transfer efficiency without additional modifications, such as input enhancements.

Table 3 Success rate on test scenario with partially assembled PCB

Method	Transferred policy	Trained policy
SAC-SV	**81%**	**100%**
SAC-CV	58%	**100%**
SAC-MV	18%	99%
SAC-EV	0%	73%

All methods trained from scratch with visual information acquired from tool cameras achieved a 100% success rate. Nevertheless, in terms of transferability, SAC-SV performed best among them, indicating that our method is robust to visual perturbation

Bold entries highlight the best results in a single column to improve visualization for the reader. In the column's top part, positions with the highest success rates are bolded. The positions with the lowest assembly times are highlighted in the bottom part

3.5 Real-World Applicability

We analysed the vision-based RL methods presented in Section 3.3 in terms of their usage on real-world production lines. In production scenarios, multiple PCBs are packed into a single panel. Hence, the policy that operates on the visual information acquired from the external camera placed in the robot workspace poorly scales to the PCB not used during the training. Each PCB from the panel would require a separate camera, and it is challenging to acquire similar images across all vision sensors. RL algorithms are known to be sensitive to changes in the observation space [35, 36]. A slight difference in the background causes a significant decrease in performance. We confirmed this statement through experiments in which SAC-EV for other PCBs of the panel failed to insert any electronic part.

In contrast, methods that use visual information acquired from the vision sensor attached to the robot tool provide a stable background independent of the location of the PCB. The results presented in the previous sections showed that our method is the most robust to pose and visual disturbances. Therefore, SAC-SV meets the requirements of applicability in real-world production applications.

4 Conclusion

This paper presents a stereo-view RL approach for the electronic part insertion task. We chose Soft Actor-Critic as our core algorithm. Our experiments show that vision-driven RL methods, combined with a compliance control system, can assemble delicate components that are vulnerable to applied forces. We evaluated the performance of the RL agent with different visual information sources used in existing works.

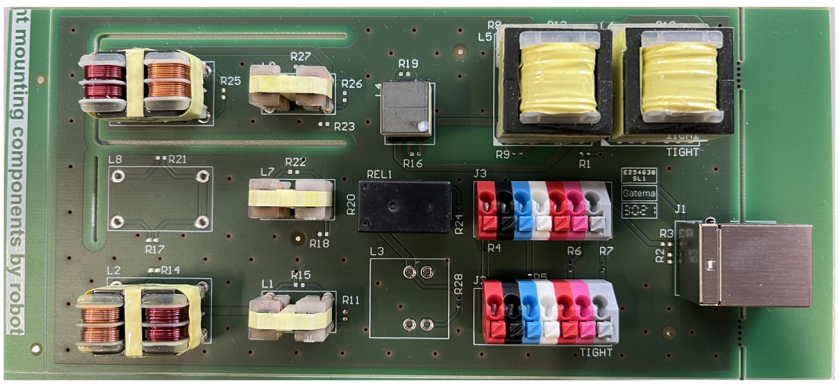


Fig. 7 Partially assembled PCB used for the experiments. The surrounding elements significantly modify the view observed by the policy

All of them achieved a success rate of more than 95% for test scenarios with low values of applied disturbances of initial position and rotation on the Z-axis. However, when the disturbances' limits were increased, our method outperformed the others in terms of the percentage of tasks successfully completed.

We also showed that our stereo vision system attached to the robot's end-effector to acquire visual information and the method to extract features from the stereoview observation space is more suitable for real production scenarios than the configuration with an external camera. In the case of the method that uses an external camera for image acquisition, the camera field of view is set up only for one PCB from the panel. This method achieved a high success rate only on the PCB used for training and failed to complete the insertion task on other PCBs of the panel. The dual-camera vision system focusses only on the assembly place and its surroundings.

Following the experiments that evaluated performance against pose disturbances, we also examined the transferability of the policy trained on the empty PCB to the partially assembled PCB. The results showed that our method achieved the best performance during these test scenarios, with a success rate of 81%. The performance of mono-view methods dropped significantly when the view was partially occluded. Moreover, in these experiments, we presented that processing stereo-view observation space by separate neural networks shows relatively high efficiency.

The advantages of RL algorithms for the assembly of electronic parts have also been demonstrated by comparison with conventional methods such as straight-down insertion, random search, and spiral search. Compared to the one presented in this work, conventional methods were not robust to the perturbations applied to the initial position and rotation over the z-axis. It should be noted that our technique could be also combined with other RL approaches for continuous control, such as TD3 or PPO. However, SAC is known for its sample efficiency, which is a desirable feature for real-world tasks.

In future research, we will verify our method in production lines and gather more information on overall performance and robustness. We are also planning to work on the problem of fast adaptation to the new tasks, defined as adjusting trained policy to new products on the production lines and new robotic stations placed in the factory without training from scratch. Achieving adaptability to unseen environment variants could significantly increase the usability of RL-based methods on high-mix, low-volume production lines where products are constantly changing. We believe that the RL-based method will replace conventional methods on production lines.

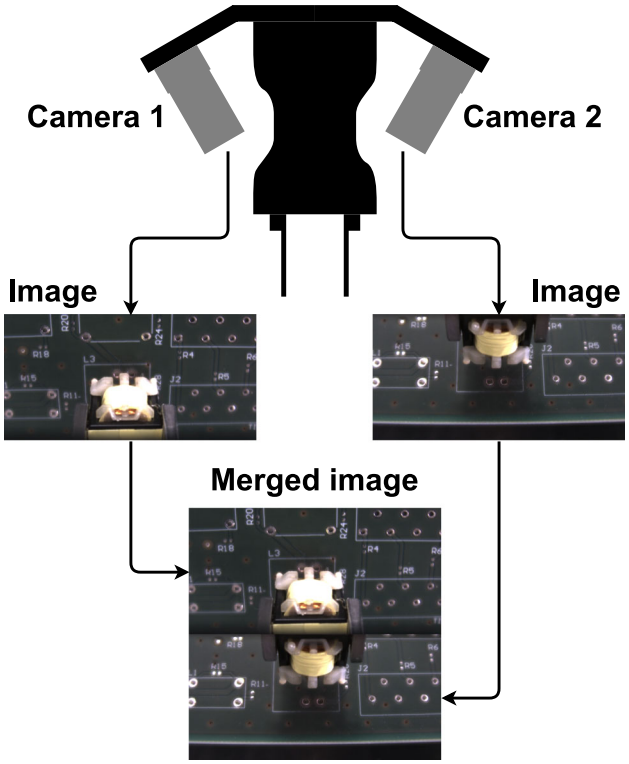


Fig. 8 Concept scheme of the image concatenation process for SAC-CV method

Appendix A: Experimental Setup - Additional Information

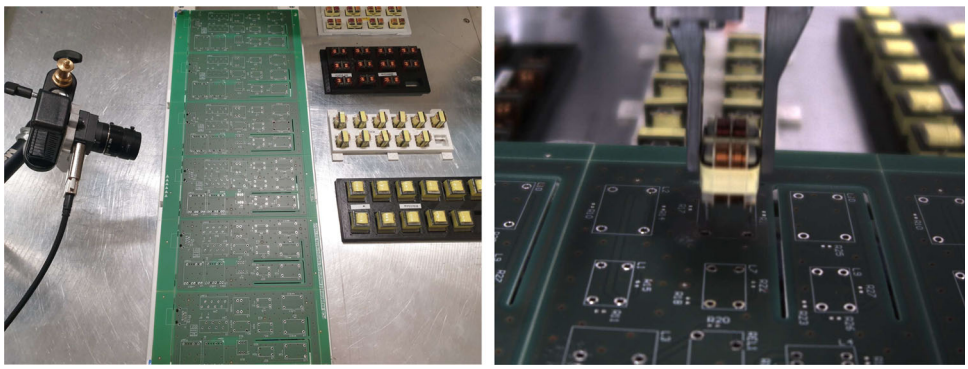
A.1 Combined View Setup

In the SAC-CV experiments, we followed the procedure described by [6] to obtain the combined view as a visual observation. Images acquired from two cameras attached on the robot's end-effector are merged into an output image of 1024×1024 pixels and then down-sampled to 128×128 pixels. Camera 1 points to the left side of the gripper and camera 2 points to the right. Such an approach provides a 360-degree-like vision in one image. The concept scheme is presented in Fig. 8

A.2 External Camera Setup

For the SAC-EC experiments, we placed an external camera in the robot's workspace (the detailed setup is presented in Fig. 9a). We set up the camera to get the field of view on the one PCB from the panel. The image acquired from this setup is illustrated in Fig. 9b (Tables 4, 5 and 6).

Appendix B: Additional Results



(a) (b)

Fig. 9 The test bed for external camera experiments. The test-bed for external camera experiments. The camera was set up in the robot's workspace to achieve the field of view focusing on the one PCB from the panel

Table 4 Summary of the experiments comparing the performance of the methods for the component type 1b (Fig. 6b)

Methods	Without noise	Position		Orientation		Mix	
		±2 mm	±5 mm	±3°	±7°	±2 mm, ±3°	±5 mm, ±7°
	Assembly success rate						
SAC-SV (ours)	**100%**	**100%**	59%	**100%**	**95%**	**100%**	45%
SAC-CV	**100%**	**100%**	57%	**100%**	93%	**100%**	44%
SAC-MV	**100%**	**100%**	48%	**100%**	**95%**	99%	39%
SAC-EV	94%	86%	**68%**	93%	77%	92%	**48%**
SAC-Force	0%	0%	0%	0%	0%	0%	0%
Straight down	86%	0%	0%	67%	26%	0%	0%
Random search	90%	26%	3%	75%	22%	14%	0%
Spiral search	98%	25%	1%	68%	26%	17%	0%
	Assembly time [s]						
SAC-SV (ours)	1.90±0.19	**1.86**±0.2	3.53±1.60	1.89±0.19	2.45±1.02	1.93±0.23	4.23±1.60
SAC-CV	**1.82**±0.19	1.87±0.19	3.19±1.37	**1.85**±0.18	**2.27**±0.85	**1.92**±0.25	**3.79**±1.29
SAC-MV	2.03±0.48	2.12±0.63	3.37±1.34	2.22±0.91	2.91±1.23	2.22±0.71	3.95±1.46
SAC-EV	2.07±0.28	2.20±0.62	**3.05**±1.05	2.18±0.25	2.97±1.14	2.57±0.88	4.16±1.09
SAC-Force	3.27±1.51	4.62±0.81	5.44±1.30	3.88±1.52	4.53±1.11	4.85±0.48	–
Straight down	1.77±0.48	–	–	2.18±0.86	1.93±0.69	–	–
Random search	1.84±0.33	2.73±0.9	3.29±1.16	2.33±0.87	2.05±0.66	3.04±0.81	–
Spiral search	1.99±0.67	2.81±1.21	3.71±0.0	2.29±0.94	2.46±1.07	2.76±1.17	–

The 100 trials of the insertion validated each method. The second table shows the average assembly times with standard deviations
Bold entries highlight the best results in a single column to improve visualization for the reader. In the column's top part, positions with the highest success rates are bolded. The positions with the lowest assembly times are highlighted in the bottom part

Table 5 Summary of the experiments comparing the performance of the methods for the component type 2 (Fig. 6c)

Methods	Without noise	Position		Orientation		Mix	
		±2 mm	±5 mm	±3°	±7°	±2 mm, ±3°	±5 mm, ±7°
	Assembly success rate						
SAC-SV (ours)	**100%**	**100%**	53%	**100%**	**99%**	**100%**	58%
SAC-CV	97%	97%	50%	95%	91%	94%	43%
SAC-MV	97%	95%	58%	99%	**99%**	97%	51%
SAC-EV	**100%**	**100%**	**91%**	**100%**	**99%**	**100%**	**78%**
SAC-Force	0%	0%	0%	0%	0%	0%	0%
Straight down	79%	0%	0%	63%	28%	0%	0%
Random search	92%	18%	2%	67%	41%	13%	0%
Spiral search	90%	21%	0%	56%	26%	18%	0%
	Assembly time [s]						
SAC-SV (ours)	2.22±0.23	2.22±0.26	3.55±1.50	2.31±0.32	2.55±0.69	2.31±0.32	3.55±1.44
SAC-CV	2.44±0.73	2.29±0.65	3.50±1.38	2.52±0.73	2.84±1.01	2.54±0.79	3.58±1.32
SAC-MV	2.32±0.68	2.43±0.76	3.33±1.34	2.39±0.61	2.60±0.80	2.48±0.75	3.63±1.30
SAC-EV	**2.02**±0.20	**2.01**±0.21	**2.93**±1.11	**2.05**±0.25	**2.35**±0.58	**2.03**±0.29	**3.31**±1.23
SAC-Force	–	–	–	–	–	–	–
Straight down	2.46±0.77	–	–	2.82±0.83	2.71±0.71	–	–
Random search	2.41±0.63	3.16±0.92	4.12±0.30	2.81±0.89	2.82±0.66	3.00±0.88	–
Spiral search	3.08±1.04	3.68±0.91	–	3.19±0.97	3.14±1.08	3.62±0.76	–

The 100 trials of the insertion validated each method. The second table shows the average assembly times with standard deviations
Bold entries highlight the best results in a single column to improve visualization for the reader. In the column's top part, positions with the highest success rates are bolded. The positions with the lowest assembly times are highlighted in the bottom part

Table 6 Summary of the experiments comparing the performance of the methods for the component type 3 (Fig. 6d)

Methods	Without noise	Position		Orientation		Mix	
		±2 mm	±5 mm	±3°	±7°	±2 mm, ±3°	±5 mm, ±7°
	Assembly success rate						
SAC-SV (ours)	**100%**	**100%**	67%	**100%**	**100%**	**100%**	61%
SAC-CV	**100%**	**100%**	52%	**100%**	**100%**	**100%**	46%
SAC-MV	**100%**	**100%**	64%	**100%**	97%	**100%**	54%
SAC-EV	**100%**	**100%**	**83%**	**100%**	**100%**	**100%**	**75%**
SAC-Force	0%	0%	0%	0%	0%	0%	0%
Straight down	8%	0%	0%	9%	12%	0%	0%
Random search	12%	3%	3%	24%	17%	12%	0%
Spiral search	16%	3%	0%	22%	20%	7%	0%
	Assembly time [s]						
SAC-SV (ours)	1.90±0.17	1.91±0.21	3.22±1.54	1.87±0.22	2.08±0.43	1.89±0.20	3.50±1.54
SAC-CV	**1.82**±0.15	1.88±0.22	2.90±1.51	**1.81**±0.17	2.00±0.39	1.89±0.22	3.42±1.59
SAC-MV	1.83±0.18	1.92±0.22	3.14±1.50	1.93±0.25	2.28±0.88	1.93±0.25	3.45±1.55
SAC-EV	1.83±0.19	**1.82**±0.20	**2.48**±1.11	1.82±0.22	**1.97**±0.40	**1.84**±0.21	**2.88**±1.38
SAC-Force	–	–	–	–	–	–	–
Straight down	3.20±0.84	–	–	2.72±1.07	2.02±0.63	–	–
Random search	2.71±0.80	2.28±0.35	3.59±0.44	2.05±0.50	2.47±0.86	2.10±0.60	–
Spiral search	2.92±0.94	3.34±0.67	–	2.65±0.94	2.42±0.98	4.41±0.45	–

The 100 trials of the insertion validated each method. The second table shows the average assembly times with standard deviations
Bold entries highlight the best results in a single column to improve visualization for the reader. In the column's top part, positions with the highest success rates are bolded. The positions with the lowest assembly times are highlighted in the bottom part

Author Contributions All authors contributed to the proposed approach. The development of the SAC-SV algorithm was performed by G. Bartyzel. The implementation of the algorithm and the control system on the robot was performed by G. Bartyzel. The test scenarios were designed by G. Bartyzel, W. Półchłopek and D. Rzepka. The figures were prepared by G. Bartyzel. The first draft of the manuscript was written by G. Bartyzel and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The research was carried out in collaboration with the company Fitech as part of the project funded by the Polish National Centre for Research and Development. Project title: "Intelligent robot for autonomous handling of assembly of electronic components based on artificial intelligence and neural networks" and number: POIR.01.01.01-00-0123/19

Code or Data Availability Data sharing not applicable to this article.

Declarations

Ethics approval This study did not require ethics approval.

Consent to Participate Not applicable. This study did not involve human subjects.

Conflict of interest The authors declare that they have no conflict of interest.

References

1. 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 (2019)
2. Kroemer, O., Niekum, S., Konidaris, G.: A review of robot learning for manipulation: challenges, representations, and algorithms. J. Mach. Learn. Res. **22**(30), 1–82 (2021)
3. Whitney, D.E.: Force feedback control of manipulator fine motions. J. Dyn. Syst. Meas. Control. **99**(2), 91–97 (1977). https://doi.org/10.1115/1.3427095
4. Park, H., Bae, J.H., Park, J.H., Baeg, M.H., Park, J.: Intuitive peg-in-hole assembly strategy with a compliant manipulator. In: Ieee Isr 2013, pp. 1–5 (2013)
5. Huang, S., Murakami, K., Yamakawa, Y., Senoo, T., Ishikawa, M.: Fast peg-and-hole alignment using visual compliance. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 286–292 (2013)
6. Triyonoputro, J.C., Wan, W., Harada, K.: Quickly inserting pegs into uncertain holes using multi-view images and deep network trained on synthetic data. In: 2019 IEEE/RSJ International conference on Intelligent Robots and Systems (IROS), p. 5792–5799 (2019)
7. Yu, C., Cai, Z., Pham, H., Pham, Q.C.: Siamese convolutional neural network for sub-millimeter-accurate camera pose estimation and visual servoing. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 935–941 (2019)
8. Inoue, T., De Magistris, G., Munawar, A., Yokoya, T., Tachibana, R.: Deep reinforcement learning for high precision assembly tasks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 819–825 (2017)
9. Johannink, T., Bahl, S., Nair, A., Luo, J., Kumar, A., Loskyll, M., et al.: Residual reinforcement learning for robot control. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 6023–6029 (2019)
10. Luo, J., Solowjow, E., Wen, C., Ojea, J.A., Agogino, A.M., Tamar, A., et al.: Reinforcement learning on variable impedance controller for high-precision robotic assembly. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 3080–3087 (2019)
11. Schoettler, G., Nair, A., Luo, J., Bahl, S., Aparicio Ojea, J., Solowjow, E., et al.: Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5548–5555 (2020)
12. Vecerik, M., Sushkov, O., Barker, D., Rothörl, T., Hester, T., Scholz, J.: A practical approach to insertion with variable socket position using deep reinforcement learning. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 754–760 (2019)
13. Xie, L., Yu, H., Zhao, Y., Zhang, H., Zhou, Z., Wang, M., et al.: Learning to fill the seam by vision: sub-millimeter peg-in-hole on unseen shapes in real world. In: 2022 International Conference on Robotics and Automation (ICRA), pp. 2982–2988 (2022)
14. Sathirakul, K., Sturges, R.H.: Jamming conditions for multiple peg-in-hole assemblies. Robotica **16**(3), 329–345 (1998). https://doi.org/10.1017/s0263574798000393
15. Fei, Y., Zhao, X.: An assembly process modeling and analysis for robotic multiple peg-in-hole. J. Intell. Robot. Syst. **36**(2), 175–189 (2003). https://doi.org/10.1023/a:1022698606139
16. Hou, Z., Li, Z., Hsu, C., Zhang, K., Xu, J.: Fuzzy logic-driven variable time-scale prediction-based reinforcement learning for robotic multiple peg-in-hole assembly. IEEE Trans. Autom. Sci. Eng. **19**(1), 218–229 (2022). https://doi.org/10.1109/tase.2020.3024725
17. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al.: Bengio, Y., LeCun, Y., (eds.) Continuous control with deep reinforcement learning
18. Hou, Z., Dong, H., Zhang, K., Gao, Q., Chen, K., Xu, J.: Knowledge-driven deep deterministic policy gradient for robotic multiple peg-in-hole assembly tasks. In: 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 256–261 (2018)
19. Xu, J., Hou, Z., Wang, W., Xu, B., Zhang, K., Chen, K.: Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks. IEEE Transactions on Industrial Informatics **15**(3), 1658–1667 (2019). https://doi.org/10.1109/tii.2018.2868859
20. Ma, Y., Xu, D., Qin, F.: Efficient insertion control for precision assembly based on demonstration learning and reinforcement learning. IEEE Transactions on Industrial Informatics **17**(7), 4492–4502 (2021). https://doi.org/10.1109/tii.2020.3020065
21. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Dy, J., Krause, A., (eds.) Proceedings of the 35th International Conference on Machine Learning, vol. 80, pp.

1861–1870. Proceedings of Machine Learning Research. Stockholm, Sweden: Pmlr (2018)
22. Haarnoja, T., Ha, S., Zhou, A., Tan, J., Tucker, G., Levine, S.: Learning to walk via deep reinforcement learning. In: Proceedings of Robotics: Science and Systems. Freiburgim-Breisgau, Germany (2019)
23. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv:1707.06347 (2017)
24. Fujimoto, S., van Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods. In: Dy, J., Krause, A., (eds.) Proceedings of the 37th International Conference on Machine Learning. vol. 80, pp. 1587–1596. Proceedings of Machine Learning Research. Stockholm, Sweden: Pmlr (2018)
25. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. A Bradford Book, Cambridge, MA, USA (2018)
26. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Bengio, Y., LeCun, Y., (eds.) 2nd International Conference on Learning Representations (2014)
27. Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., et al.: Soft Actor-Critic algorithms and applications. arXiv
28. Villani, L., Schutter, J.D., Khatib, O.: Force control. In: Siciliano, B., (ed.) Springer Handbook of Robotics. Springer International Publishing, pp. 195–220 (2016)
29. Horgan, D., Quan, J., Budden, D., Barth-Maron, G., Hessel, M., van Hasselt, H., et al.: Distributed prioritized experience replay. In: International Conference on Learning Representations (2018)
30. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: ICML Workshop on Deep Learning for Audio, Speech and Language Processing, vol. 30 (2013)
31. Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., Fergus, R.: Improving sample efficiency in model-free reinforcement learning from images. Proceedings of the AAAI Conference on Artificial Intelligence **35**(12), 10674–10681 (2021). https://doi.org/10.1609/aaai.v35i12.17276
32. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot operating system 2: design, architecture, and uses in the wild. Science Robotics **7**(66), eabm6074 (2022). https://doi.org/10.1126/scirobotics.abm6074
33. Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., et al.: RLlib: abstractions for distributed reinforcement learning. In: International Conference on Machine Learning (ICML) (2018)
34. Marvel, J.A., Bostelman, R., Falco, J.: Multi-robot assembly strategies and metrics. ACM Comput. Surv. **51**(1) (2018). https://doi.org/10.1145/3150225
35. Huang, S.H., Papernot, N., Goodfellow, I.J., Duan, Y., Abbeel, P.: Adversarial attacks on neural network policies. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings; (2017)
36. Kos, J., Song, D.: Delving into adversarial attacks on deep policies. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings; 2017

Grzegorz Bartyzel obtained his M.Sc. in automatic control and robotics from the AGH University of Science and Technology, Kraków, Poland in 2017. Currently, he is pursuing a PhD in robotic engineering and artificial intelligence at the same university. His research topics include the use of reinforcement learning methods to adapt industrial robots to new tasks in production lines. Since 2017, he has been working as an Artificial intelligence Engineer at Fitech and Stellantis, developing algorithms for industrial robots and autonomous vehicles. His research interests include control theory, robotics and general applications of artificial intelligence with a particular emphasis on reinforcement learning.

Dominik Rzepka received his M.Sc. and Ph.D. degrees in electrical engineering from the AGH University of Science and Technology, Kraków, Poland, in 2009 and 2018, respectively. From 2007 to 2011, he was with the Wireless Sensor and Control Networks Group at AGH University of Science and Technology, where he was involved in design of the low-power algorithms for the processing of radio signals and software-defined radio. In 2011, he joined the Event-Based Control and Signal Processing Group at AGH University of Science and Technology, where he currently works on methods of signal reconstruction from event-triggered samples and on neuromorphic machine learning. In 2014-2023 he was a Visiting Student and Postdoc Researcher in the University of Manitoba, Winnipeg, Canada, and in The City College of New York, USA. Since 2015, he has been working as Signal Processing and Machine Learning Researcher at Comarch Healthcare and Fitech, developing algorithms for diagnostics and quality assurance systems. His research interests include signal processing and machine learning in biomedicine, wireless communication and industrial inspection, as well as event-based systems.

Dr Eng. Wojciech Półchłopek received his master's degree in electronics engineering in 1999 from the Faculty of Electrical Engineering, Automatics, Computer Science and Electronics at the AGH University of Science and Technology in Krakow. In 2007, he obtained a doctoral degree in technical sciences in the field of Electronics from the same faculty. From 1999 to 2011, he worked at the Department of Electronics at AGH. From 2011 to 2017, he implemented projects involving medical diagnostics devices and algorithms at iMed24 Medical Devices/Comarch Healthcare. Since 2017, he has been involved in developing algorithms in the field of image processing using AI/DSP techniques for quality control applications at Fitech. He is the author of numerous practical applications of signal processors in audio and speech signal processing, as well as medical diagnostics. He is also the author of innovative solutions in the field of artificial intelligence applications in vision systems.