Computers & Industrial Engineering 208 (2025) 111313

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

An offline reinforcement learning-based framework for proactive robot assistance in assembly task

Yingchao You, Boliang Cai, Ze Ji *

School of Engineering, Cardiff University, Queen's Buildings, The Parade, Cardiff, CF24 3AA, Wales, UK

ARTICLE INFO

Keywords:
Assembly
Human–robot collaboration
Proactive assistance
User preference
Offline reinforcement learning

ABSTRACT

Proactive robot assistance plays a critical role in human–robot collaborative assembly (HRCA), enhancing operational efficiency, product quality and workers' ergonomics. The shift toward mass personalisation in industries brings significant challenges to the collaborative robot that must quickly adapt to product changes for proactive assistance. State-of-the-art knowledge-based task planners in HRCA struggle to quickly update their knowledge to adapt to the change of new products. Different from conventional methods, this work studies learning proactive assistance by leveraging reinforcement learning (RL) to train a policy, ready to be used for robot proactive assistance planning in HRCA. To address the limitations therein, we propose an offline RL framework where a policy for proactive assistance is trained using the dataset visually extracted from human demonstrations. In particular, an RL algorithm with a conservative Q-value is utilised to train a planning policy in an actor–critic framework with carefully designed state space and reward function. The experimental results show that with only a few demonstrations performed by workers as input, the algorithm can train a policy for proactive assistance in HRCA. The assistance task provided by the robot can fully meet the task requirement and improve human assembly preference satisfaction by 47.06% compared to a static strategy.

1. Introduction

Human–robot collaboration (HRC) has been widely adopted in manufacturing for tasks such as collaborative assembly. Proactive assistance from collaborative robots is essential in these tasks, as it not only enhances safety and efficiency but also enables smoother workflows by anticipating human needs and reducing cognitive load, hence fostering a more seamless integration between human workers and robotic systems. Research by Liu, Chen, Abuduweili, and Liu (2023) demonstrates that timely proactive assistance can reduce workers' idle time and improve workflow in assembly tasks. Besides, mass personalisation (MP) is an advanced technique that allows customers to personalise their products, powered by techniques in industry 4.0 (Wang, Ma, Yang, & Wang, 2017). The transformation to MP needs a quick response to customer needs and to keep the operation efficient in manufacturing (Zhang & Ming, 2023). However, numerous personalised products present challenges to collaborative robot operation in production (Othman & Yang, 2023), where robots need to rapidly acquire skills in new product assembly operations to cope with product changes.

To realise proactive assistance in HRCA, many task-level planning systems for HRCA were proposed for generating collaborative robot actions. However, these systems typically require some form of prior knowledge about the tasks as prerequisites, formulated using techniques such as task graph (Darvish, Simetti, Mastrogiovanni, & Casalino, 2020; Lee, Behdad, Liang, & Zheng, 2022), planning domain definition language (PDDL) (Izquierdo-Badiola, Canal, Rizzo, & Alenyà, 2022) and ontology (Chang, Cho, & Choi, 2020; Umbrico, Orlandini, & Cesta, 2020). In most existing works, such task knowledge is usually pre-programmed by domain experts (Stramandinoli, Roncone, Mangin, Nori, & Scassellati, 2019). Manually specifying the task knowledge by domain experts is time-consuming and not user-friendly. Therefore, the above methods are impractical for quickly updating the assembly knowledge for new product assembly tasks.

RL-based methods allow robots to acquire skills through the exploration of the environment, which does not need to build an explicit knowledge model beforehand. RL aims to learn the optimal policy by maximising accumulated rewards through interaction with the environment. This process is usually formulated as a Markov Decision Process (MDP). RL techniques have achieved significant success in various robot applications, including autonomous navigation (Zhu & Zhang, 2021) and manipulation (Jangir, Alenya, & Torras, 2020). This work considers proactive assistance in an assembly line, where the robot

* Corresponding author.
E-mail addresses: Youy4@cardiff.ac.uk (Y. You), caib4@cardiff.ac.uk (B. Cai), JiZ1@cardiff.ac.uk (Z. Ji).

https://doi.org/10.1016/j.cie.2025.111313
Received 7 October 2024; Received in revised form 16 April 2025; Accepted 7 June 2025
Available online 26 July 2025

provides assistance by handing over the right parts to human operators according to the users' requirements. For the assembly process, due to its stochastic decision-making nature, it can be modelled as an MDP (Biemer & Cooper, 2023). Since RL can learn through trial and error and improve through exploration, we believe RL has the potential to offer an alternative solution that enables robots to learn proactive assistance faster and more effectively compared to existing methods. Therefore, the core problem this work aims to address is: *How can we design an RL-based method for robots to acquire assembly knowledge and provide proactive handover to human workers?*

It is non-trivial to design an RL-based method for proactive assistance, because of the following limitations:

1. Online interaction: In general, typically, RL needs to interact online with the environment for exploration. However, due to the unpredictable nature of random exploration in RL methods, collecting training data and interacting with the environment in the presence of humans is risky and expensive. Thus, deploying standard RL solutions in the context of HRCA is impractical.
2. Uncertainty in workers' operation: Workers' preference on assembly operation varies based on human natural tendencies and ergonomic needs, and meeting the preference of workers is necessary for user satisfaction improvement (Aheleroff, Huang, Xu, & Zhong, 2022). However, the uncertainty of workers' operation brings challenges to the RL for stable performance on the proactive assistance.
3. Task requirement. The assembly sequence of the product must adhere to its task requirements. Ensuring that handed-over parts meet these requirements is crucial, as it guarantees that parts are handed over in the correct order. This prevents disruptions and rework, ultimately enhancing productivity (Othman & Yang, 2023).

To address the above limitations, this work proposes an offline RL-based framework designed to quickly train a policy for task-level proactive assistance in HRCA by learning from human assembly demonstrations. The demonstration of the human assembly process is full of implicit task knowledge as well as human operation preference, which is an ideal resource for collaborative robot learning. By collecting human demonstrations and employing the visual extraction method, a static dataset is created for offline RL policy training. The observation space, action space, and rewards function for the offline RL are also designed correspondingly for stable performance. The proposed framework allows the robot to learn to provide assistance that meets both task requirements and human preferences with only a few demonstrations by humans.

The contributions of this paper are summarised as follows:

1. A novel offline RL-based framework is proposed for learning robot assistance in HRCA tasks. This approach eliminates the need for risky and expensive interactive data collection for policy training in HRCA.
2. An offline RL-based HRCA system is developed for robot assistance learning that requires only a few demonstrations and a relatively short training period.
3. The system is validated through a real-world HRC experiment involving multiple participants. The results demonstrates that the learned robot assistance can fully meet assembly task requirements and accommodate 47.06% more user operation preferences compared with the static strategy.

2. Related work

2.1. Task planning model for collaborative robots in assembly tasks

HRC integrates the strengths of robots and human operators, emerging as a pivotal model in manufacturing. It demonstrates significant potential for applications across various domains, such as welding (Liu & Bao, 2024; Liu, Zheng, & Bao, 2023). Many task-planning systems for collaborative robots were proposed to generate collaborative robot motions (Simões, Pinto, Santos, Pinheiro, & Romero, 2022). Most of the task planning methods for collaborative robots belong to the knowledge-based model in the literature, such as the and/or graph, ontology, and Planning Domain Definition Language (PDDL). Darvish et al. (2020) proposed a FlexHRC+ architecture for robots to support human operators for manufacturing tasks. It integrates a specified and/or graph to model action knowledge using first-order logic. Chang et al. (2020) proposed an ontology-based model to model general knowledge about agents' actions, domain knowledge, and environment for collaborative robots. Izquierdo-Badiola et al. (2022) built an HRC planning system using PDDL for replanning in failure anticipation when the state of the human operator changes. Such task knowledge models are usually pre-programmed by domain experts (Stramandinoli et al., 2019). Manually specifying task knowledge by domain experts is time-consuming and unintuitive, impeding the development of HRC in the industry. In the context of MP, rapidly updating robotic task planning systems to meet the assembly needs arising from product changes is very challenging.

2.2. Reinforcement learning for collaborative robot

RL-based methods enable robots to acquire skills by interacting with environments, which can avoid manually designing the task planning model. In HRCA, researchers have applied RL to address various challenges. For example, Yu, Huang, and Chang (2020) used RL to optimise working sequences, thereby improving the efficiency of human–robot teams. Li, Zheng, Yin, Pang, and Huo (2023) used deep RL for motion planning to prevent collisions between robotic arms and human operators, ensuring safety. These studies conducted the learning process in simulated environments, without direct interaction with human operators. However, online RL methods are impractical for collaborative robot task planning due to the high risks and costs associated with collecting training data through direct human interaction (Li, Hu, Zhou, & Pham, 2023). Consequently, online RL methods are not suitable for quickly enabling a collaborative robot for a new product in manufacturing processes.

We aim to explore offline RL for the task planning of collaborative robots to avoid direct interaction with human users in the training process. Human expert demonstrations of the assembly process are ideal for a collaborative robot to learn how to act as a human-like assistant because it can provide the data for offline RL training. Observing the human assembly process, our method can quickly model the knowledge required for the assembly task and provide appropriate assistance actions accordingly.

2.3. Handover methods for collaborative robots

Handover is a crucial skill for robots in the context of HRC, and many researchers have studied this issue from different perspectives. For example, a visual and haptic perception combined with a control method has been proposed to adapt robot behaviour and grip force to human actions, ensuring safe and smooth handovers (Costanzo, De Maria, & Natale, 2021). Cini, Banfi, Ciuti, Craighero, and Controzzi (2021) investigated the impact of the timing of a robot's handover intention signals on the tasks being performed by human operators. Their findings indicate that the timing of these signals significantly affects the performance of human–robot teams. Peternel, Kim, Babič, and Ajoudani (2017) proposed a control method based on a dynamic human model to calculate the optimal handover position, achieving handovers that meet ergonomic conditions.

The studies above focus mainly on the action level. However, for multi-step assembly tasks, handover at the symbolic level presents additional challenges. Meeting the task requirements of handed-over

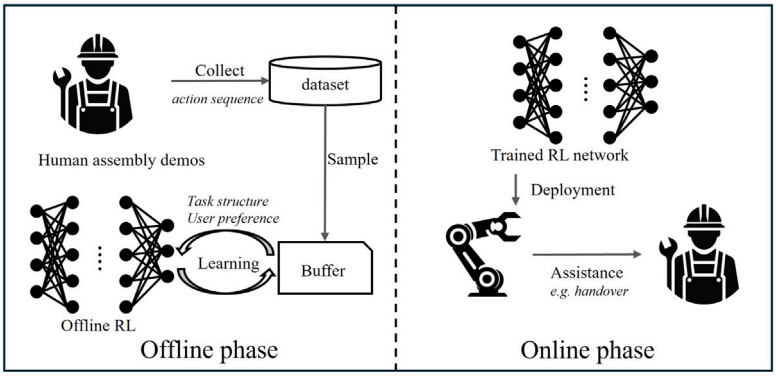

Fig. 1. The framework of learning from human demonstration for proactive assistance.

parts is essential, as it ensures that parts are provided in the correct sequence. If the parts handed over by the robot do not meet the current task requirements, it can disrupt the assembly process and reduce efficiency (Othman & Yang, 2023). Additionally, considering human sequence preferences is equally important. Product assembly may follow various sequences. Skilled workers often have a preferred order based on their natural tendencies. For instance, some workers prefer to complete tasks requiring the same tool consecutively to avoid switching tools mid-process. If the parts handed over by the robot align with the worker's preferences, it can significantly enhance worker satisfaction and efficiency (Aheleroff et al., 2022). Therefore, our proposed method aims to ensure both the task requirements and human sequence preferences in the assistance provided by assistive robots.

3. Problem formulation

In a robot–human assembly unit, we assume the presence of a robot r and an assembly worker h. The robot assists the worker in completing the assembly of product P, which consists of N parts. The assembly sequence of the product follows the product's **task requirements**, which refer to the logical relationships between assembly actions as dictated by the geometric information of the product. The robot assists the worker by handing over parts, reducing the time the worker spends retrieving parts and improving assembly efficiency. The robot should proactively hand over parts to the worker based on the current assembly status, ensuring that the action aligns with the worker's assembly preferences and the product's task requirements. To enable the robot's proactive assistance in new assembly tasks, the primary goal is to develop a policy that predicts the worker's next assembly part at each step, based on the current product assembly status. This policy should satisfy both the product's task requirements and the worker's assembly preferences. Therefore, the objective of the policy is to maximise the assistance score $J(\pi)$.

$$\max_{\pi} J(\pi) = \max_{\pi} \sum_{n=1}^{N} s_t + s_p \tag{1}$$

where s_t represents the score for meeting the product's task requirements at step n, and s_p represents the score for meeting the worker's assembly preferences at step n.

4. Method

This paper aims to propose an offline RL-based framework where robots can quickly learn to provide proactive assistance to assembly workers by observing human assembly processes. In the rest of the section, we first introduce the overall framework, followed by the details of the offline RL algorithm in the framework used to address the proactive assistance learning problem.

4.1. Framework

The general framework we propose for quick learning from demonstration for proactive assistance in assembly tasks is depicted in Fig. 1. This framework is divided into two phases: the offline phase and the online phase. It enables the robot to learn from human demonstrations offline and use the acquired policy to assist workers in assembly tasks online.

Offline Phase: When faced with a new assembly task or a product for which the robot lacks relevant knowledge, a worker demonstrates the assembly process to the robot, and the demonstration should meet assembly requirements and the worker's preferences. The worker's actions and product states during the assembly process are captured using visual recognition methods. The action sequences and states of product are then stored in a dataset. An offline RL algorithm is trained using samples from this dataset. Through this process, the task structure information and human sequence preferences are implicitly learned. The offline reinforcement learning algorithm is deployed to maximise the cumulative reward, resulting in an optimal policy that can be used to plan the robot's assistance actions during collaborative assembly. It is important to note that the action recognition algorithm is not the focus of this paper.

Online Phase: The robot uses the learned policy to assist the worker in assembling the product. During execution, the product assembly state is identified using visual methods. The trained policy is then deployed on the robot, enabling it to proactively control the robot and provide assistance to the assembly worker based on the recognised assembly state.

4.2. Proactive assistance modelling as a Markov decision process

To model the human assembly operation process of a product, we formulate it using an MDP, defined by a tuple $M = (S, A^H, P_a, \gamma, R_a)$. S is the state space of the product assembly status. $A^h \in A$ is a set of discrete operational actions performed by the human operator on the product. $A^r \in A$ denotes the assistance action performed by the robot. $P_a(s'|s,a)$ is the transition probability from the state $s \in S$ to state $s' \in S$ under action $a \in A$. $\gamma \in [0,1)$ is the discount factor. R_a is the reward after the robot assists humans with the corresponding assembly tasks.

The human demonstrations of assembly are denoted by: $D = \{\eta_1, \eta_2, \ldots, \eta_I\}$. where $\eta_i, i \in I$ is one assembly process of human demonstration. η_i consists of the assembly states and the human actions as $\eta_i = \{(s_{i,1}, a_{i,1}), (s_{i,2}, a_{i,2}), \ldots, (s_{i,k}, a_{i,k})\}$. Based on that, a policy π for proactive assistance in HRCA will be developed. We assume that robot assistance is handing over the parts according to the human user's preference and production requirement. The robot deploys policy π to predict the next human operator action $A^h \sim \pi(\cdot, S)$ according to the observed assembly state. Based on the prediction, the robot can execute an action to hand over the corresponding part to the human operator proactively.

4.2.1. Observation space

The observation space for the robot is defined by the state of the parts, represented as O. The dimension of the observation space depends on the number of parts N that make up the product:

$$O = [o_1, \dots, o_n, \dots, o_N] \tag{2}$$

where o_n represents the status of part p_n. The value of o_n can take one of three possible states:

0: to be assembled,
1: being assembled,
2: be assembled.

The observation state of desktop case assembly (used in the validation experiment) is defined by a 1×8 vector, where element i corresponds to the state of component i. $[0, 0, 0, 0, 0, 0, 0, 0]$ indicates that all components are in the "to be assembled" state.

4.2.2. Action space

The action space consists of the robot's assistance actions in the assembly process, denoted as A_r. The action space is defined as a discrete set of possible actions from which the agent selects one at each decision point. Each a_n does not represent a binary value independently; rather, the entire vector represents a mutually exclusive choice of a single action:

$$A_r = [a_1, \dots, a_n, \dots, a_N, a_h, a_{idle}] \tag{3}$$

where a_n is the action that picks up the corresponding parts or tools p_n, a_h is the action that hands over the object in hand to the worker, and a_{idle} means the agent is idle.

The action space of desktop case assembly (used in the validation experiment) is defined as $A_r = [a_1, \dots, a_8, a_h, a_{idle}]$, where a_1 to a_8 represent actions such as picking up the motherboard, CPU, cooler, GPU, memory card, hard disk, power supply, and cover.

4.2.3. Reward function

The performance of RL heavily depends on the design of the reward function. In this work, the reward function is constructed to guide the selection of actions that complete the assembly task while satisfying the product's task requirements. The first component of the reward function focuses on achieving the shared goal of the agents, completing the assembly, where a positive reward is given upon completion. Additionally, to ensure that the parts are passed adhere to the task requirements, a negative reward is assigned if these requirements are not met. The reward function does not incentivise actions that align with worker preferences, as the training data is derived from human demonstrations. In this case, actions that meet the worker's preferences already have a higher probability in $P_a(s'|s, a)$.

The reward function $r(s, a)$ is designed as follows:

$$r(s,a) = \begin{cases} r_c & \text{if an assembly task is done.} \\ r_w & \text{if an assembly step does not meet the assembly} \\ & \text{task requirement.} \\ r_s & \text{if a component is assembled successfully and meets} \\ & \text{the assembly task requirement.} \end{cases} \tag{4}$$

where r_c is a completion reward if an assembly task is done, r_w is a negative reward if the agent executes an action that does not meet the assembly task requirement, r_s denotes the intermediate reward granted when a component is successfully assembled and the corresponding assembly task requirement is satisfied.

In the case of the PC desktop, $r_c = 50$, highlighting the significant importance of completing the entire assembly task. A penalty of $r_w = -2$ is applied to discourage invalid actions. Additionally, $r_s = 5$ serves as an intermediate reward, encouraging incremental progress and reducing the likelihood of premature termination.

4.3. Offline RL approach for proactive assistance

To learn proactive assistance from human assembly demonstration data, we adopt an offline RL framework, which relies on static datasets rather than direct online interaction with the environment for policy training. This approach avoids the risky and expensive human-in-the-loop interactions with online environments.

We introduce an offline actor–critic training framework to address the MDP problem described above. In this setup, the actor generates a possible action distribution based on the current state, while the critic estimates the value function to evaluate the action taken by the actor under the given policy. The actor and critic are both parameterised by the deep neural networks.

4.3.1. Data collection and prepossessing

Initially, a static dataset D for policy training needs to be constructed, and the worker's assembly demonstration is recorded using cameras. Inspired by advances in machine vision, we apply action/object recognition techniques to preprocess the video stream, extracting the worker's action a, the assembly state s at step t, and the assembly state s' when the action is done. Finally, the reward r is calculated using (4), and the tuple (s, a, s', r) is stored in dataset D. This process is repeated until the product is assembled completely. During policy training, mini-batches are randomly sampled from dataset D to efficiently utilise the data for training the policy and Q-value network. The details of the algorithm are shown in the algorithm 1.

4.3.2. Solving overestimation of Q-value in critic

Overestimation of Q-values is a common issue when estimating Q-values by the critic, especially in offline RL (Meng, Gorbet, & Kulić, 2021). This occurs because the model may tend to overestimate the Q-values of actions that are infrequently represented in the dataset. Inaccurate Q-value estimation can negatively impact the performance of the proactive assistance policy. Recent work (Kumar, Zhou, Tucker, & Levine, 2020) has adopted a conservative Q-value estimation method to avoid overestimation. Therefore, in this study, we use the loss function for the critic by incorporating a conservative regularisation term in addition to the standard Bellman loss, which prevents overestimation of out-of-distribution actions and stops the actor from exploiting such actions. Additionally, the critic employs a dual Q-network setup, where two Q-networks are used for Q-value estimation. By taking the minimum of the two estimated Q-values, the method helps reduce the positive bias in Q-value estimates, improving the accuracy and stability of the learning process. The critic updates its Q-function by minimising the $\mathcal{L}_{\text{critic}}$.

$$\begin{aligned} \mathcal{L}_{\text{Critic}} = & \underbrace{\frac{1}{2}\mathbb{E}_{(s,a,r,s')\sim D}\left[\left(Q(s,a) - \left(r + \gamma \max_{a'} Q(s',a')\right)\right)^2\right]}_{\text{Bellman loss}} \\ & + \alpha \underbrace{\left(\mathbb{E}_{s\sim D, a\sim\pi_\theta}[Q(s,a)] - \mathbb{E}_{s\sim D, a\sim\mu}[Q(s,a)]\right)}_{\text{Conservative regularisation term}} \end{aligned} \tag{5}$$

where $Q(s, a)$ is the Q-function being learned, r is the observed reward, and γ is the discount factor, μ is the behaviour policy that generated the dataset D. $\mathbb{E}_{s\sim D, a\sim\pi_\theta}[Q(s, a)]$ denotes the expected Q value of the learned policy π_θ; $\mathbb{E}_{s\sim D, a\sim\mu}[Q(s, a)]$ denotes the expected Q-value under the behaviour policy μ; α is a regularisation coefficient.

4.3.3. Actor update

The actor π_ϕ can be updated using gradient descent by maximising the expected Q-value predicted by the critic π_θ, while also promoting exploration by maximising the policy's entropy, as done in the soft actor–critic framework (Haarnoja et al., 2018). This combination helps

balance the exploitation of high-value actions with the exploration of new actions, improving overall policy performance.

$$\mathcal{L}_{\text{Actor}} = \mathbb{E}_{s\sim D}\left[\mathbb{E}_{a\sim\pi_\phi(a|s)}\left[\alpha \log \pi_\phi(a|s) - Q_\theta(s,a)\right]\right] \tag{6}$$

where α is the temperature parameter that controls exploration and exploitation, $\pi_\phi(a|s)$ is the policy learned by the actor, and $Q_\theta(s,a)$ is the Q-value learned by the critic, which includes conservative regularisation.

Algorithm 1 Offline RL method for quick learning from demonstration for proactive assistance in an assembly task

Input: Offline dataset $\mathcal{D}$, learning rates η_θ, η_ϕ, discount factor γ, temperature parameter α, soft update factor τ
Output: Optimised policy $\pi_\phi(a|s)$ and Q-networks $Q_\theta^{(1)}(s,a)$, $Q_\theta^{(2)}(s,a)$

Initialise Dual Q-networks (Critic) $Q_\theta^{(1)}(s,a)$ and $Q_\theta^{(2)}(s,a)$ with random parameters $\theta^{(1)}$, $\theta^{(2)}$
Initialise Dual target Q-networks $Q_{\theta_{\text{target}}}^{(1)}(s,a)$, $Q_{\theta_{\text{target}}}^{(2)}(s,a)$ with $\theta_{\text{target}}^{(i)} \leftarrow \theta^{(i)}$
Initialise policy network (Actor) $\pi_\phi(a|s)$ with parameters ϕ
for each step t do **do**
  Sample a mini-batch of transitions (s,a,r,s') from dataset $\mathcal{D}$
  Critic Update:
  Compute target Q-value: $y(r,s') = r+\gamma(1-d)\min_{i=1,2} Q_{\theta_{\text{target}}}^{(i)}(s',a')$
  for each $i \in \{1,2\}$ **do**
    compute critic losses using equation (5)
    Update critic parameters: $\theta^{(i)} \leftarrow \theta^{(i)} - \eta_\theta \nabla_\theta \mathcal{L}_{\text{Critic}}^{(i)}(\theta)$
  end for
  Actor Update:
  Sample actions $a \sim \pi_\phi(a|s)$ from current policy
  Minimise actor loss using equation (6)
  Update actor parameters: $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathcal{L}_{\text{Actor}}(\phi)$
  Target Networks Update:
  Update target networks: $\theta_{\text{target}}^{(i)} \leftarrow \tau\theta^{(i)} + (1-\tau)\theta_{\text{target}}^{(i)}$
end for

5. Case study and evaluation

In this section, a prototype system for the HRCA of desktop cases is developed based on the proposed framework. Then, a performance evaluation with multiple participants is conducted for the system in terms of task requirements and human preference. Lastly, we perform a qualitative analysis by comparing our method with the state-of-the-art assistive robot task-planning techniques in robots, highlighting their advantages and disadvantages.

5.1. The HRCA prototype system of desktop cases

This section provides an overview of the system pipeline and the technical details of the developed HRCA system for the desktop case assembly.

5.1.1. The system structure

The proposed Robot Operating System (ROS)-based system comprises two main components, which are shown in Fig. 2: the offline phase for data collection and policy training, and the online phase for desktop case HRCA execution.

Offline Phase: In the offline phase, several demonstrations of the assembly process from workers are first collected using an RGB-D camera. In these demonstrations, to simplify the task of vision-based state observation, ArUco markers (Kam, Yu, & Wong, 2018) are attached to the parts for object tracking. The worker's actions and assembly states are identified using an action recognition module and an object tracking module (You, Ji, Yang, & Liu, 2022), forming a dataset. This dataset, containing user actions and assembly states, is used to train the offline RL algorithm.

On the other hand, instead of deploying the trained policy in the real-world environment directly, we employ a simulation-based approach first for effective policy evaluation. The simulation environment is developed with a simple graphical user interface, based on the GYM API for policy training and evaluation, as shown in Fig. 3. The trained policy with the best reward is saved for real-world deployment.

Online Phase: During the assembly execution, real-time action recognition and object tracking methods are employed, using RGB-D data of the assembly scene as input, to identify the assembly state and track the human operator. The identified assembly state is then input into the trained policy to compute recommended actions. By tracking the human operator, the system calculates the distance between the operator and the part to be handed over. If the part's position exceeds the user's pick threshold, the robot inquires whether assistance is needed. Upon receiving a positive response, the robot executes the assistance action. To ensure safety in the human–robot assembly cell, an OctoMap-based module is used for collision avoidance and motion planning. After each action, the system continuously monitors the assembly state and provides proactive assistance to the human operator until the assembly is completed. The hardware used in the system includes a KUKA iiwa LBR robot, a Realsense D435 camera, and an RTX 3080 GPU for algorithm training.

5.1.2. Experimental subject

In the system, we use a desktop PC case as the assembly subject, a typical product requiring manual assembly in manufacturing, as shown in Fig. 5(a). The task requirements for the desktop case are represented in an and-or graph, which defines constraints on the assembly sequence, as illustrated in Fig. 5(b).

5.1.3. Collision avoidance

OctoMap is a 3D occupancy grid to model arbitrary environments in real time. We use point cloud data of the product area to build a collision avoidance system based on OctoMap (Hornung, Wurm, Bennewitz, Stachniss, & Burgard, 2013), which creates an obstacle map for the motion planning area. Motion planning algorithms, specifically the RRT* algorithm (Noreen, Khan, & Habib, 2016), are employed to compute the execution path, which is then carried out by the robot controller. The experiment environment and the corresponding OctoMap is shown in Fig. 4.

5.2. System evaluation

To validate the effectiveness of the system, we conducted a proof-of-concept assembly experiment involving multiple participants, and then evaluated our proposed method regarding the alignment of human sequence preference and task requirements.

5.2.1. Experiment description

To validate our proposed framework, we propose the following three hypotheses:

1. Does robot assistance reduce the energy or effort, referred to as physical exertion, required by the human operator to complete the assembly task?
2. Does the robot's assistance action meet the current assembly task requirements?
3. Does the robot's assistance action align with the user's preferred assembly sequence?

To address the three hypotheses, we established three research groups to perform a human–robot collaborative desktop assembly task. The group utilising our method serves as the experimental group, with two control groups included for comparison. In the experimental

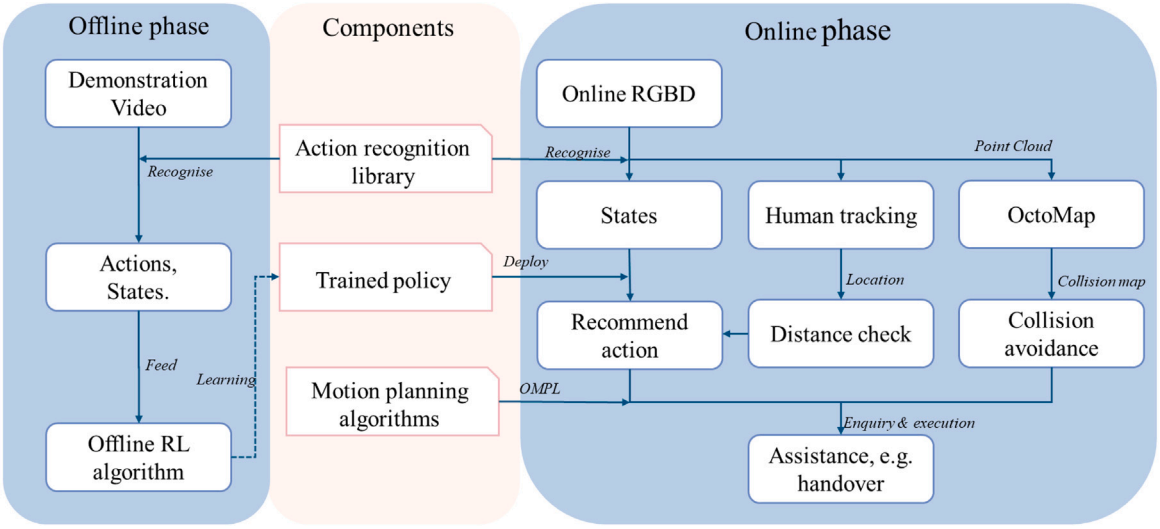

Fig. 2. The system pipeline of the prototype desktop case HCRA system.

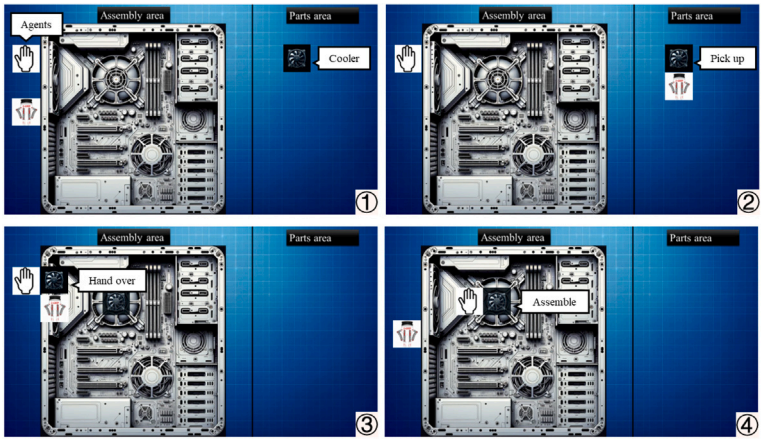

Fig. 3. To efficiently evaluate the trained policy without directly testing it in the real environment, a desktop assembly simulation environment is developed. This environment simulates the desktop assembly task at a symbolic level. It includes an assembly area and a part area, where agents are a worker and a robot. The robot is controlled with the trained policy. The action space comprises actions such as picking up a part, assembling a part, and handing over a part. The reward is defined in Eq. (4). In the figure, an example of this process is shown in which the robot picks up the cooler and hands it over to the worker, who then assembles it.

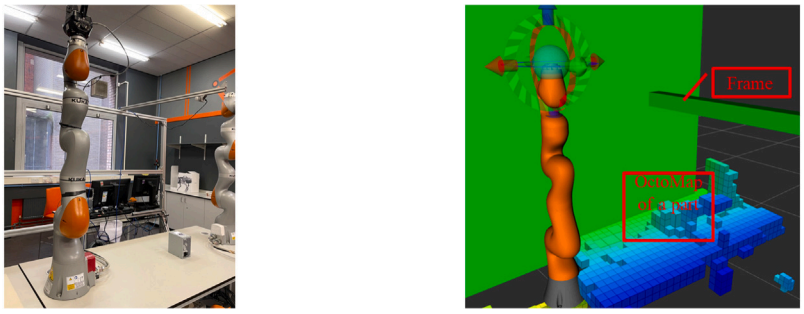

Fig. 4. Octomap-based obstacle avoidance module for the HRC system safety is illustrated in the figure. (a) The experimental human–robot assembly cell consists of a work desk, a KUKA robot, and an RGB-D camera mounted on the frame to capture the environment's point cloud. (b) The virtual assembly environment, features static obstacles (green), OctoMap-based dynamic obstacles, and a robot capable of performing safe motion planning within the environment.

group, the collaborative robot is controlled using our proposed method. Initially, we recorded 3 demonstrations of each participant's assembly processes. Participants were allowed to assemble the product according to their preferences,resulting in an average of 51.5 collected transitions per participant, subsequently used for offline RL policy training. An example is shown in Fig. 6. The algorithm is trained over 200k steps using the collected data. The policy that achieved the best results in the simulation environment will be saved and then used to determine the robot's assistance actions in the real experiment.

The two control groups are as follows:

Control Group 1: The robot's actions are hard-coded, assisting participants by following a fixed assembly sequence, which is a common method for robot task planning models in industrial settings.

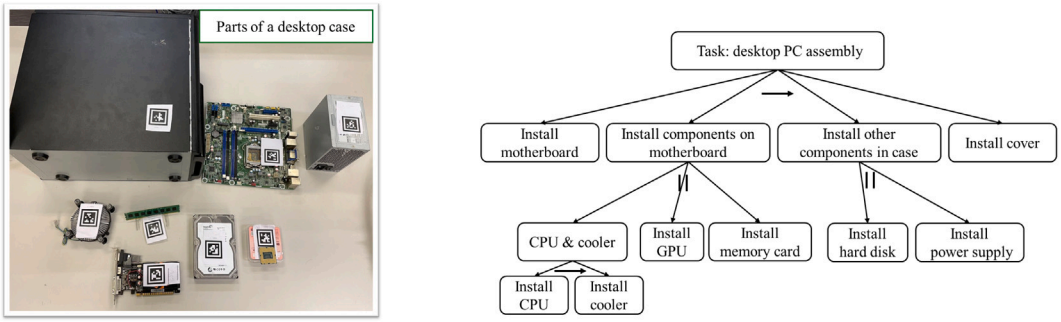

Fig. 5. The figures illustrate the desktop used in the experiment. (a) The desktop case, with each part labelled using ArUco markers. The desktop consists of nine parts: the case, motherboard, CPU, cooler, GPU, memory card, hard disk, power supply, and cover. (b) The and-or graph of the desktop case defines the task requirements. The "→" indicates a sequential relationship, while "||" defines a parallel relationship between tasks.

Fig. 6. The top-down view illustrates the demonstration of the desktop assembly process by a human operator. A visual method is used for tracking both the hands and the parts. Action and state recognition are performed using the method in You et al. (2022).

Control Group 2: The subjects assemble the desktop case without robotic assistance, simulating a manual assembly scenario.

5.2.2. Experiment metrics

Questionnaire is a subjective way to evaluate the performance of the policy in the real-world experiment. We invited 8 subjects to participate in our experiment to collect experimental data. Each subject participated in all three groups of experiments. They were blind to the type of group they were assigned to. Participants were asked to complete three questionnaire for each group when all experiments were completed. Each question in the questionnaire is designed to address one of the proposed hypotheses. We conducted a statistical analysis of the collected results to validate the three hypotheses proposed. In the questionnaire,

- Question 1 assesses the physical exertion required to complete the tasks in the three groups, using the Modified Borg Scale (Wilson & Jones, 1989), with scores ranging from 0 to 10.
- Question 2 evaluates the number of robot assistance actions that align with the human operators' sequence preferences in the experimental group and Control Group 1.
- Question 3 measures the number of actions that meet the task requirements in the experimental group and Control Group 1.

We also use the best reward as a key metric for evaluating the policy's performance. Specifically, the highest attainable best reward is 90, achieved under conditions where $r_c = 50$, indicating the successful assembly of 8 components ($r_s = 8 \times 5$), with no invalid actions ($r_w = 0$). This sum yields a total reward of 90. Given this known optimal performance, the convergence progress and efficiency of the policy can be effectively observed and assessed.

5.2.3. Experiment result

For each participant, we trained the policy using their demonstration data over 200k steps. The trained policy was tested at every 1000 steps in the simulation environment based on the best reward. The regularisation coefficient α is a crucial hyperparameter in the proposed offline RL algorithm, balancing Q-learning and conservatism, directly impacting the algorithm's performance. To determine the optimal value of α, we conducted an experiment to identify which value leads to the best performance in the proactive assistance task. The values of α tested were $[0.01, 0.04, 0.08, 0.1, 0.3, 0.5, 0.7, 0.8, 1, 10]$.

Table 1
The best reward achieved by the offline RL algorithm is evaluated for different values of α. The "best reward" refers to the highest reward the algorithm attains during the 200 training epochs. The "epoch" refers to the point at which the algorithm first attains its highest reward during training.

The value of α	Best reward	Epoch
0.01	5	26
0.04	5	11
0.08	5	11
0.1	90	20
0.3	90	9
0.5	90	35
0.7	90	18
0.8	25	14
1	25	14
5	10	4
10	10	2

Tested in the simulation environment, the variation in the best reward of the trained offline RL policy for different values of α is illustrated in Fig. 8 and Table 1. As shown, the values $\alpha = [0.1, 0.3, 0.5, 0.7]$ achieved the highest reward of 90, with $\alpha = 0.3$ converging the quickest at step 9k. When α is greater than 0.7, the algorithm becomes overly conservative, favouring only the most frequently observed actions from the dataset while ignoring potentially valuable but underexplored actions. Conversely, when α is less than 0.1, the model tends to overestimate Q-values for actions that are not well-represented in the dataset, particularly for out-of-distribution actions. Based on these results, both overestimated and underestimated Q-values can negatively affect performance in the proactive assistance task.

The well-trained policy was also tested in the real environment, and the process is shown in Fig. 7. The experimental results from the questionnaire are discussed as follows:

1. **Physical Exertion**: The physical exertion of the eight participants across the three groups is depicted in the box plot in Fig. 9(a). The control group 2 required the most physical exertion, with an average value of approximately 7.6. The physical exertion required in the experimental group and control group 1 were lower than that in control group 2, indicating that robotic assistance reduces the physical exertion needed to complete the assembly tasks. This supports Hypothesis 1.

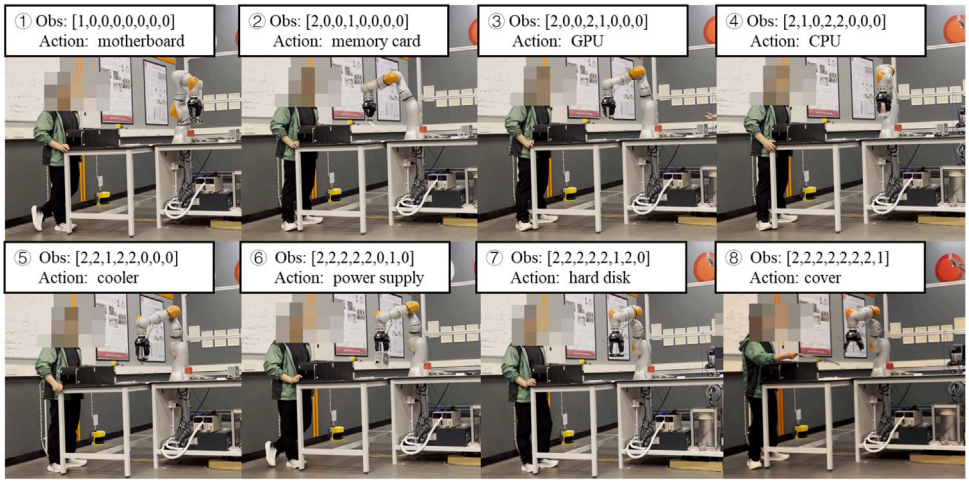

Fig. 7. This figure demonstrates the process of the robot assisting the worker in assembling the desktop case. It includes both the observations and the agent's actions. The initial state observed by the camera is [0,0,0,0,0,0,0,0], indicating that all eight parts are yet to be assembled. The robot first picks up the motherboard and hands it to the operator. Subsequently, the robot sequentially hands over the memory card, GPU, CPU, cooler, power supply, and hard disk to the operator. Finally, since the cover is close to the operator, he picks it up and completes the assembly.

2. **Task Requirement**: In the experimental group, the robot provided a total of 56 assistance actions for the eight subjects, all of which met the task requirements for the current assembly state. The 100% accuracy in meeting task requirements demonstrates that the proposed method reliably provides assistance actions that fulfil the task needs, confirming Hypothesis 2.
3. **Users' Sequence Preference**: The number of assistance actions that meet the users' sequence preferences in the experimental group and control group 1 is shown in the box plot in Fig. 9(b). The average number of actions meeting the users' sequence preferences is 6.25 in the experimental group and 4.25 in control group 1. The users' sequence preference in the experimental group is 47.06% higher than that in control group 1, indicating that the proposed method significantly aligns with the users' preferred assembly sequence, confirming Hypothesis 3.

5.2.4. System efficiency

We measured the average training time over 200k steps for the model of eight subjects, which was 28.07 min. This indicates that the robot can quickly learn the assistance actions in a relatively short time. Additionally, we measured the model's computation time during the execution phase, with an average time of 0.00425 s. This demonstrates the model's capability for real-time decision-making.

5.3. Comparison study of offline reinforcement learning algorithms

This section aims to evaluate the performance of various reinforcement learning algorithms in proactive assistance learning by comparing them with our proposed method. For this comparison, we selected state-of-the-art algorithms that are well-suited for discrete offline reinforcement learning. The baseline algorithms include Critic Regularised Regression (CRR) (Wang et al., 2020), Batch-Constrained Deep Q-Learning (BCQ) (Fujimoto, Conti, Ghavamzadeh, & Pineau, 2019), Behavioural Cloning (BC) (Hussein, Gaber, Elyan, & Jayne, 2017) and Decision Transformer (DT) (Chen et al., 2021).

The best reward serves as a critical metric for assessing the effectiveness of the algorithms in proactive assistance learning, while the epoch to reach the highest reward measures the convergence rate of each algorithm. These two metrics were chosen as key indicators to evaluate the algorithms' performance.

The results presented in Table 2 show that our algorithms, BCQ and DT, outperform the other methods in terms of overall performance,

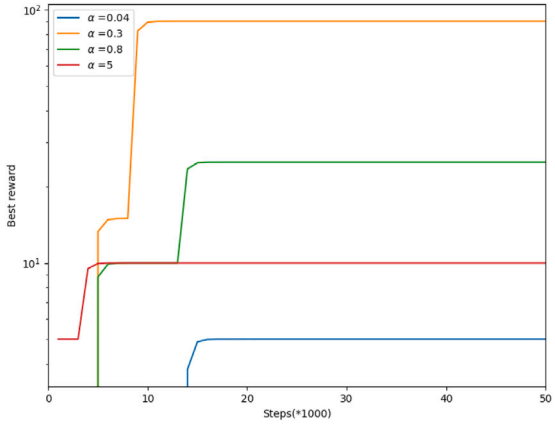

Fig. 8. The variation in the best reward of the trained offline RL policy for different values of α.

Table 2
The outcomes of the comparative analysis of offline reinforcement learning algorithms.

Algorithms	Best reward	Epoch
Ours	90	9
CRR	89.6	168
BCQ	90	14
BC	10	24
DT	90	4

each achieving the maximum reward of 90. Among them, DT demonstrates the fastest convergence, reaching the optimal reward within just 4 epochs. In contrast, CRR achieves a near-optimal reward of 89.6, but requires 168 epochs, indicating lower learning efficiency. BC performs significantly worse, with a reward of only 10, underscoring its limitations in handling complex tasks.

5.4. The qualitative analysis of the proposed and state-of-the-art methods

A qualitative analysis of the proposed and state-of-the-art methods for task planning in HRCA is conducted concerning modelling methods, expertise, efficiency, and alignment with task requirements and human user preferences. The analysis results are shown in Table 3.

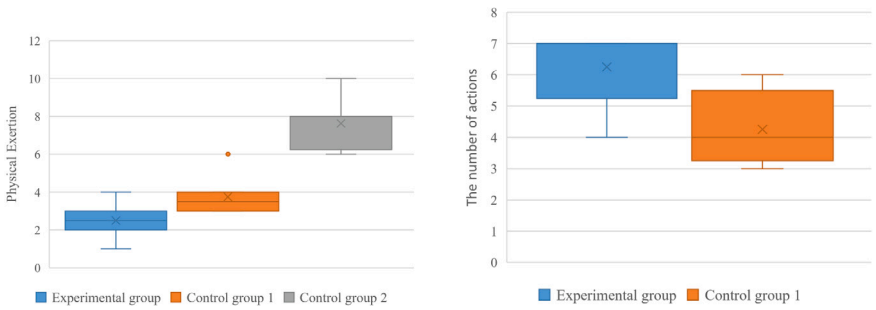

Fig. 9. (a) The results of participants' physical exertion across three experiments, as reported in the questionnaires. (b) The number of actions that align with user sequence preferences, comparing the experimental group and the control group, as gathered from the questionnaires.

Table 3
The qualitative analysis results by comparing the proposed and the state-of-the-art methods.

Methods	Description	Efficiency	Expertise	Task requirement	User preference
Ours	A method for modelling robot assistance actions based on offline RL by observing worker assembly operations.	Several minutes to several hours (for both the demonstration and algorithm training time)	N	Y	Y
Audio-guided method (Wang, Li, Chen, Diekel, & Jia, 2019)	A method for providing assistance actions using inverse reinforcement learning based on voice commands.	Several minutes to several hours (for both the audio demonstration and algorithm training time)	N	Y	Y
CAD-based method (Schirmer, Kranz, Rose, Schmitt, & Kaupp, 2023)	A method for generating assembly sequence information using CAD models.	Several hours to several days (primarily for CAD modelling)	Y	Y	N
PDDL-based method (Jiang, Zhang, Khandelwal, & Stone, 2019)	A task planning method modelled using PDDL.	Several hours to several days (primarily for PDDL modelling)	Y	Y	N

The most similar work to ours in the literature is by Wang et al. (2019), who proposed an audio-guided method for assistance action task planning model based on inverse reinforcement learning. Like our method, this approach requires a short modelling time. However, it necessitates voice prompts to inform the robot of the current assembly state and actions, which is not the most natural workflow and adds an extra burden on the assembly worker. The CAD-based modelling method generates assembly sequences based on the CAD model of the product. This method is effective only when a CAD model is available. However, creating CAD models is time-consuming, potentially taking several hours or even days, depending on the complexity of the product. The PDDL-based modelling method is a task-planning approach. Its drawback is the significant amount of time required to construct PDDL models. Both CAD-based and PDDL-based methods require specialised domain knowledge, which is not typically possessed by assembly workers. All modelling methods consider the ability to meet task requirements. However, when it comes to aligning with user sequence preferences, only our method and the one proposed by Wang et al. (2019) fulfil this criterion.

5.5. Discussion and future work

The real-world experiment validated the proposed method's effectiveness in alleviating fatigue, meeting task requirements, and enhancing workers' operational preferences, confirming three hypotheses. These findings demonstrate the proposed method's ability to improve user experience in HRC. Additionally, the analysis of system efficiency confirmed the method's performance in training efficiency and real-time decision-making, showcasing its capability in implementation decisions. A comparative study of offline RL algorithms highlighted the performance of different approaches in proactive assistance and convergence speed. Finally, we conducted a qualitative analysis of the mechanism, efficiency, and performance of similar methods.

Nevertheless, the proposed method has certain limitations: Validation during the training process of offline reinforcement learning remains a challenge. Due to the inability to interact with real-world environments, validation becomes difficult. This study adopted a compromise by constructing a simulation environment to validate the model. In future work, we will explore alternative evaluation approaches for offline RL methods to improve the practicality of the proposed method. Furthermore, we intend to expand the sample size and incorporate a broader range of participant characteristics to gain deeper insights into factors that enhance user experience in HRC.

Another limitation is that our model is customised for each worker, with training data derived exclusively from the demonstration of that worker. This constraint limits the model's generalisation ability across different individuals. In future work, we will explore solutions to address this limitation.

6. Conclusion

We proposed a novel offline RL-based framework for enabling robot assistance in HRCA by learning from human demonstrations, eliminating the need for risky and expensive real-time interactions for policy training. Based on this framework, we developed a ROS-based HRCA system that can quickly learn robot assistance actions while ensuring safety. To validate our proposed framework, we designed an experiment demonstrating the advantages of learning robot assistance actions based on human demonstrations. The results demonstrate the benefits of our method, where only a few human demonstrations are required,

featuring a quick training process. Also, the learned actions fully meet both assembly task requirements and user sequence preferences. The benefits show that our proposed method provides a feasible path for learning assistive robots in HRCA.

In future work, we plan to integrate alternative object tracking and action recognition methods into our framework to enhance its flexibility. Additionally, we will explore combining the advantages of CAD-based and voice-based methods with our proposed approach to develop a robust method for assistive robot task planning models.

CRediT authorship contribution statement

Yingchao You: Writing – original draft, Validation, Methodology, Investigation, Conceptualization. **Boliang Cai:** Validation. **Ze Ji:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Conceptualization.

Acknowledgement

Yingchao You, Boliang Cai thank the China Scholarship Council for providing scholarships for their Ph.D. programmes (No. 202006020046, 202106710026).

Data availability

Data will be made available on request.

References

Aheleroff, S., Huang, H., Xu, X., & Zhong, R. Y. (2022). Toward sustainability and resilience with industry 4.0 and industry 5.0. *Frontiers in Manufacturing Technology*, *2*, Article 951643, Publisher: Frontiers.

Biemer, C. F., & Cooper, S. (2023). Level assembly as a markov decision process. arXiv preprint arXiv:2304.13922.

Chang, D. S., Cho, G. H., & Choi, Y. S. (2020). Ontology-based knowledge model for human-robot interactive services. In *Proceedings of the 35th annual ACM symposium on applied computing* (pp. 2029–2038).

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., et al. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, *34*, 15084–15097.

Cini, F., Banfi, T., Ciuti, G., Craighero, L., & Controzzi, M. (2021). The relevance of signal timing in human-robot collaborative manipulation. *Science Robotics*, *6*(58), eabg1308, Publisher: American Association for the Advancement of Science.

Costanzo, M., De Maria, G., & Natale, C. (2021). Handover control for human-robot and robot-robot collaboration. *Frontiers in Robotics and AI*, *8*, Article 672995, Publisher: Frontiers Media SA.

Darvish, K., Simetti, E., Mastrogiovanni, F., & Casalino, G. (2020). A hierarchical architecture for human–robot cooperation processes. *IEEE Transactions on Robotics*, *37*(2), 567–586, Publisher: IEEE.

Fujimoto, S., Conti, E., Ghavamzadeh, M., & Pineau, J. (2019). Benchmarking batch deep reinforcement learning algorithms. http://dx.doi.org/10.48550/arXiv.1910.01708, URL: http://arxiv.org/abs/1910.01708, arXiv:1910.01708.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., et al. (2018). Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, *34*, 189–206, Publisher: Springer.

Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys*, *50*(2), 21:1–21:35. http://dx.doi.org/10.1145/3054912.

Izquierdo-Badiola, S., Canal, G., Rizzo, C., & Alenyà, G. (2022). Improved task planning through failure anticipation in human-robot collaboration. In *2022 international conference on robotics and automation* (pp. 7875–7880). IEEE.

Jangir, R., Alenya, G., & Torras, C. (2020). Dynamic cloth manipulation with deep reinforcement learning. In *2020 IEEE international conference on robotics and automation* (pp. 4630–4636). IEEE.

Jiang, Y.-q., Zhang, S.-q., Khandelwal, P., & Stone, P. (2019). Task planning in robotics: an empirical comparison of pddl-and asp-based systems. *Frontiers of Information Technology & Electronic Engineering*, *20*, 363–373, Publisher: Springer.

Kam, H. C., Yu, Y. K., & Wong, K. H. (2018). An improvement on aruco marker for pose tracking using kalman filter. In *2018 19th IEEE/ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing* (pp. 65–69). IEEE.

Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, *33*, 1179–1191.

Lee, M.-L., Behdad, S., Liang, X., & Zheng, M. (2022). Task allocation and planning for product disassembly with human–robot collaboration. *Robotics and Computer-Integrated Manufacturing*, *76*, Article 102306. http://dx.doi.org/10.1016/j.rcim.2021.102306, URL: https://www.sciencedirect.com/science/article/pii/S0736584521001861.

Li, W., Hu, Y., Zhou, Y., & Pham, D. T. (2023). Safe human–robot collaboration for industrial settings: a survey. *Journal of Intelligent Manufacturing*, 1–27, Publisher: Springer.

Li, C., Zheng, P., Yin, Y., Pang, Y. M., & Huo, S. (2023). An AR-assisted deep reinforcement learning-based approach towards mutual-cognitive safe human-robot interaction. *Robotics and Computer-Integrated Manufacturing*, *80*, Article 102471. http://dx.doi.org/10.1016/j.rcim.2022.102471, URL: https://www.sciencedirect.com/science/article/pii/S0736584522001533.

Liu, T., & Bao, J. (2024). A novel period-sensitive LSTM for laser welding quality prediction. *IEEE Transactions on Industrial Informatics*, 1–9. http://dx.doi.org/10.1109/TII.2024.3463706, URL: https://ieeexplore.ieee.org/abstract/document/10716249/authors#authors.

Liu, R., Chen, R., Abuduweili, A., & Liu, C. (2023). Proactive human-robot co-assembly: Leveraging human intention prediction and robust safe control. In *2023 IEEE conference on control technology and applications* (pp. 339–345). http://dx.doi.org/10.1109/CCTA54093.2023.10252579.

Liu, T., Zheng, P., & Bao, J. (2023). Deep learning-based welding image recognition: A comprehensive review. *Journal of Manufacturing Systems*, *68*, 601–625. http://dx.doi.org/10.1016/j.jmsy.2023.05.026, URL: https://www.sciencedirect.com/science/article/pii/S0278612523001036.

Meng, L., Gorbet, R., & Kulić, D. (2021). The effect of multi-step methods on overestimation in deep reinforcement learning. In *2020 25th international conference on pattern recognition* (pp. 347–353). IEEE.

Noreen, I., Khan, A., & Habib, Z. (2016). A comparison of RRT, RRT* and RRT*-smart path planning algorithms. *International Journal of Computer Science and Network Security (IJCSNS)*, *16*(10), 20, Publisher: International Journal of Computer Science and Network Security.

Othman, U., & Yang, E. (2023). Human–robot collaborations in smart manufacturing environments: review and outlook. *Sensors*, *23*(12), 5663, Publisher: MDPI.

Peternel, L., Kim, W., Babič, J., & Ajoudani, A. (2017). Towards ergonomic control of human-robot co-manipulation and handover. In *2017 IEEE-RAS 17th international conference on humanoid robotics (humanoids)* (pp. 55–60). http://dx.doi.org/10.1109/HUMANOIDS.2017.8239537.

Schirmer, F., Kranz, P., Rose, C. G., Schmitt, J., & Kaupp, T. (2023). Holistic assembly planning framework for dynamic human-robot collaboration. In *International conference on intelligent autonomous systems* (pp. 215–227). Springer.

Simões, A. C., Pinto, A., Santos, J., Pinheiro, S., & Romero, D. (2022). Designing human-robot collaboration (HRC) workspaces in industrial settings: A systematic literature review. *Journal of Manufacturing Systems*, *62*, 28–43, Publisher: Elsevier.

Stramandinoli, F., Roncone, A., Mangin, O., Nori, F., & Scassellati, B. (2019). An affordance-based action planner for on-line and concurrent human-robot collaborative assembly. In *2nd ICRA international workshop on computational models of affordance in robotics*.

Umbrico, A., Orlandini, A., & Cesta, A. (2020). An ontology for human-robot collaboration. *Procedia CIRP*, *93*, 1097–1102, Publisher: Elsevier.

Wang, W., Li, R., Chen, Y., Diekel, Z. M., & Jia, Y. (2019). Facilitating human-robot collaborative tasks by teaching-learning-collaboration from human demonstrations. *IEEE Transactions on Automation Science and Engineering*, *16*(2), 640–653. http://dx.doi.org/10.1109/TASE.2018.2840345.

Wang, Y., Ma, H.-S., Yang, J.-H., & Wang, K.-S. (2017). Industry 4.0: a way from mass customization to mass personalization production. *Advances in Manufacturing*, *5*(4), 311–320, Publisher: Springer.

Wang, Z., Novikov, A., Zolna, K., Merel, J. S., Springenberg, J. T., Reed, S. E., et al. (2020). Critic regularized regression. *Advances in Neural Information Processing Systems*, *33*, 7768–7778, URL: https://proceedings.neurips.cc/paper/2020/hash/588cb956d6bbe67078f29f8de420a13d-Abstract.html?ref=https://githubhelp.com.

Wilson, R. C., & Jones, P. (1989). A comparison of the visual analogue scale and modified borg scale for the measurement of dyspnoea during exercise. *Clinical Science*, *76*(3), 277–282, Publisher: Portland Press Ltd..

You, Y., Ji, Z., Yang, X., & Liu, Y. (2022). From human-human collaboration to human-robot collaboration: automated generation of assembly task knowledge model. In *2022 27th international conference on automation and computing* (pp. 1–6). IEEE.

Yu, T., Huang, J., & Chang, Q. (2020). Mastering the working sequence in human-robot collaborative assembly based on reinforcement learning. *IEEE ACCESS*, *8*, 163868–163877. http://dx.doi.org/10.1109/ACCESS.2020.3021904.

Zhang, X., & Ming, X. (2023). A smart system in manufacturing with mass personalization (S-MMP) for blueprint and scenario driven by industrial model transformation. *Journal of Intelligent Manufacturing*, *34*(4), 1875–1893, Publisher: Springer.

Zhu, K., & Zhang, T. (2021). Deep reinforcement learning based mobile robot navigation: A review. *Tsinghua Science and Technology*, *26*(5), 674–691, Publisher: TUP.